Commodore INFO

Comdex Las Vegas

Basic cursus

C-128 voor wie?

ONAFHANKELIJK BLAD VOOR COMMODORE GEBRUIKERS PRIJS ƒ 6,50/Bfr. 130 JAARGANG 2, No.10, 1985

LISTINGS

CHECKSUM
GALOP 64
HANDWERK 64
KLUTSKWIJT 64
RATTLESNAKE
S.I.T.
DEMO SIT
TELEFOON
LOTTO 64
ADRESBESTAND

C-16 SPECIAL
9 pagina's aandacht voor de C-16 met o.a. nieuwe C-16 Checksum en Visilist

Utilities

Soundmon

Veel listings

Machinetaal-64

Champ/Spritemachine

Muziek uit de SID-chip

Superpokes voor spelfanaten

Kortingsbon voor **TAKS 64** nieuwe versie voor belastingaangifte 1985

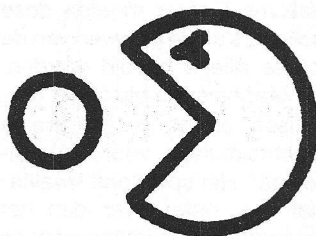
Redactioneel

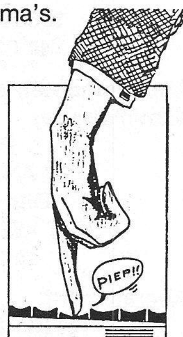
De decembermaand is traditioneel een topper voor de detailhandel. Dat geldt natuurlijk ook voor de computer- en softwarewinkels en het ziet er ook naar uit dat zij op een redelijk tevredenstemmend koopseizoen zullen kunnen terugkijken. Er zal geen hallelujahstemming zijn, maar we weten allemaal dat die tijden voor de computerbranche enigszins voorbij zijn.
Op zich hoeft niemand daar dramatisch over te doen, met het volwassen worden van een industrie komt er na vele woelingen uiteindelijk een zekere rust op de markt. Daarbij hoort ook dat er nog fikse klappen zullen vallen links en rechts en het moet U dan ook niet verbazen als er in het komende jaar berichten zullen verschijnen over in moeilijkheden verkerende bedrijven. De Amerikanen hebben een mooi woord voor dergelijke bewegingen, zij spreken van een "shake out", waarbij de sterksten overblijven. Computers zullen er echter altijd blijven en iedereen kan er ook vrij zeker van zijn dat hij voor zijn hobby steeds voldoende hardware, periferieken en software zal kunnen blijven vinden.
Een duidelijke winnaar wat betreft type of systeem lijkt er voor dit jaar nog niet uit de bus gekomen te zijn. Daarvoor moet eerst alle rook van het seizoen opgetrokken zijn. Toch is er bij de computeraars zélf al wel een duidelijke groep winnaars en wel in een heel onverwachte hoek: de C-16 bezitters. Door een grote winkelorganisatie in ons land zijn in de laatste weken namelijk niet minder dan 10.000 C-16 computers verkocht en dat heeft een hernieuwde interesse in deze eigenlijk voortdurend miskende machine tot stand gebracht.
Commodore-Info kan daar natuurlijk niet bij achterblijven en we willen zelfs het voortouw nemen bij deze herwaardering. Vandaar dat in dit nummer opnieuw uitgebreid aandacht aan de C-16 besteed is. Jan Bodzinga heeft zijn door alle C-64 gebruikers zo gewaardeerde Checksum en Visilist programma's speciaal voor de C- 16 herschreven, zodat deze belangrijke hulpprogramma's, die het leven van de hobby-programmeur veraangenamen, nu ook voor deze machine beschikbaar zijn.
Wij wensen onze lezers prettige feestdagen en een voorspoedig begin van 1986. Met een nieuwe jaargang Commodore-Info zullen wij het onze doen om daaraan bij te dragen.

L.Sala

Inhoud van dit nummer







Listingrubriek PRINT-OUT
Ook deze keer weer een groot aantal flitsende listings:

COMMODORE-INFO
Jaargang 2, no. 10, 1985

Uitgave:
Sala Communications

Uitgever:
Drs. J. Taverne

Marketing/Beurzen:
Ing. V. Sala

Redaktieadres:
Postbus 112
1260 AC Blaricum
☎ 02152-65695

Abonnementen en administratie:
Postbus 5570
1007 AN Amsterdam
Den Texstraat 5a
1017 XW Amsterdam
☎ 020-273198

Abonnement:
f 55,– of Bfr. 1100 per jaar
(10 nummers)

Betaling op Giro 158591 tnv. SAC/COMMODORE-INFO Blaricum of in België op Bank BBL nr. 310050602562, vermeld SAC/COMMODORE-INFO. Oude nummers à *f* 6,75 alleen bij vooruitbetaling op giro 1585491.

REDAKTIE
Ir. L. Sala hoofdredacteur
J. Bodzinga adj. hoofdredacteur
K. van der Vlies reportage
R. van den Heuvel listings
B. Munniksma onderwijs
Sj. Bakker machinetaal

ART DEPARTMENT
Omslagontwerp::
Ben van Mierlo

Strip:
Bert Tier

Illustraties:
Ben van Mierlo
Ymmot

Redactiesecretariaat:
J. Emmelot

ISSN: 0169-3085
NDB Zoeterwoude

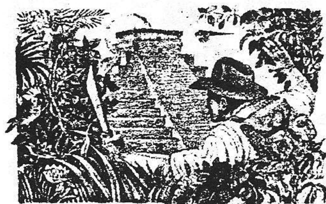
Druk:
Verweij, Mijdrecht
NDB, Zoeterwoude

Distributie:
in Nederland Betapress/Gilze
in België AMP/Brussel

nieuws

Lucas filmgames

De bekende Science Fiction filmregiseur George Lucas (o.a. Star Wars) heeft zich nu ook op de computerspelletjesmarkt gestort. Dat er een groot vertrouwen in het succes van Lucas' ideeën bestaat bleek al toen de beide eerste spelletjes klaar waren. Zij werden meteen per telefoon uit de bedrijfscomputer gekraakt.



Momenteel zijn twee tamelijk prijzige (c.a. *f* 140,=) diskettes voor de C-64 verkrijgbaar. In 'Rescue on Fractalus' kunt u op deze ongastvrije planeet een aantal gestrande astronauten gaan redden. M.b.v. de hypermoderne ruimtejager Valkyrie Fighter moeten deze collega's uit de klauwen van de lokale Aliens bevrijd worden. De graphics zijn bijzonder realistisch en er is een fraaie vluchtsimulator voor de ruimtejager. Het spel komt kwalitatief veel beter over dan het doorsnee ruimtewezentjes knallen.
De tweede titel is het spel **Ballblazer** voor C-64. Zoals de naam al enigszins aangeeft gaat het hier allerminst om een statisch spelletje en de aktie spuit dan ook van het scherm. Het betreft in feite een soort 'ruimtehockey' waarbij ook nog het doel voortdurend verandert.

Hollanditis

Een van de best getimede produkten van de laatste tijd is de nieuwe Topsoft produktie **Hollanditis.**
Zo vlak voor de nieuwe Tweede Kamerverkiezingen, net na het petitionnement en midden in de discussies over de kruisraketten lanceert succes-programmeur John 'DR.J' Vanderaart zijn laatste, op deze Nederlandse koorts geënte, adventure, met als brandende centrale vraag: wél of geen kruisraketten?
Hollanditis, een zgn. grafisch adventure geeft de speler de gelegenheid om zélf een antwoord te geven op deze vraag. In de hoofdrol assisteert Heinrich Glühwein, Zwitsers privedetective, de speler in het vergaren van geheime documenten, microfilms en belastend matriaal. Daartoe wordt Heinrich over de hele wereld gestuurd, van Moskou tot London, van Rotterdam tot New York, op zoek naar verborgen gegevens. Gegevens die een uiteindelijke plaatsing van de 48 kruisraketten in Nederland kunnen bespoedigen of... voorkomen. En daar zit nu meteen het leuke van Hollanditis: zowel pro- als anti-kruisraketten spelers kunnen hun doel bereiken en daarmee winnen. Het programma zelf neemt geen stelling. Als je voor plaatsing bent, goed. Dan moet je er voor zorgen dat ze alle 48 geplaatst worden om te winnen. Ben je tegen, dan moet je plaatsing, koste wat het kost, zien te voorkomen. En daartoe heb je 1000 uur, een budget van 25.000 gulden, een hand vol commando's en je eigen vindingrijkheid tot je beschikking. Wat meteen opvalt in het spel is de grafische verzorging. De vele exotische lokaties waar je als speler terechtkomt zijn als high- resolution graphics een van de betere voorbeelden die je op de 64 kunt aantreffen. Het avontuur wordt gedeeltelijk met het geven van tekstcommando's en gedeeltelijk met het kiezen van functietoetsen gespeeld. Daardoor ontstaat

er geen irritatie meer over het steeds maar intypen van terugkerende commando's.
Voor de oplettende speler zijn er veel grapjes in verwerkt, die ook alleen maar door intensief spelen ontdekt kunnen worden. Er is zelfs aan gedacht dat de doorsnee speler, en daar reken ik mezelf ook toe, geen benul heeft van het politieke effect van de opgespoorde gegevens. Aan wie moet je die microfilms bijvoorbeeld weer doorspelen om wel-of-niet-plaatsing te bespoedigen? Daarom is in het programma een apparaatje 'ingebouwd', de zgn. F.A.I.T. A.C.C.O.M-.P.L.I. Waar deze afkorting precies voor staat, moet je zelf in het spel maar uitvinden.
De band van Hollanditis bevat ook nog een bonusprogramma: **Radeloos**, een spin-off van het succesvolle **Eindeloos**. In dit programma raakt onze held Heinrich Glühwein verdwaald in een doolhof op zoek naar de 48 kruisraketten, wwar hij zich alleen met aktie, schieten en andere spannende aktiviteiten uit weet te redden.

Een ongeluk komt nooit alleen...

VRIJDAG DE DERTIENDE...

Van Domark
Dit is het nieuwste spel van Domark, dat de spelletjes "A View to a Kill", "Codename Mat II", and "Eureka" ook heeft gemaakt. *De vakantie van de kampeerders bij Crystal Lake wordt wreed verstoord wanneer één van de kampeerders, Jason, verdrinkt. De verbitterde moeder van Jason, die de kampeerders daarvan de schuld geeft neemt wraak. Ze vermoord alle kampeerders op één meisje na, die de moeder van Jason weet te doden.*

Het meisje, de enige overlevende, drijft dan in een bootje naar het midden van het meer, waar ineens Jason uit het water rijst om wraak te nemen op het meisje dat zijn moeder vermoord heeft...!

Jason's geest dwaalt echter nog steeds rond, wanneer je als onschuldige computeraar op vakantie bent in Crystal Lake.

De bedoeling is een veilige plaats te vinden zodat Jason je niet kan vinden en om je vrienden te overtuigen hetzelfde te doen. In dit spel zul je Jason moeten vinden die in eerste instantie onherkenbaar is, en pas opvallen als hij aanvalt. Voordat je aangevallen wordt krijg je een gelukkig waarschuwing zodat je toch de kans krijgt, hem te vinden.

Maar pas op..., Jason wacht op geen man, vrouw, of kind!

De medekampeerders hebben elk hun eigen persoonlijkheid. Zij reageren, tijdens het spel, naar hun eigen kracht en paniekgrenzen, dus rustig blijven. Gedurende het spel zijn er 10 wapens van verschillende kracht, maar het hoofddoel blijft om de onbekende Jason bekend te maken en te elimineren.
Prijs ongeveer *f* 48,–.

Philips Modem

EC 123 voor home- en personal computergebruikers.

Onlangs is een nieuw modem geïtroduceerd door de datacommunicatie-groep van Philips Telecommunicatie en Informatiesystemen te Den Haag. Het is de EC 123, mede door zijn lage prijs niet alleen aantrekkelijk voor toepassing bij home- en personal computers, maar ook voor de professionele markt.

De EC 123 is een single-board modem die gebruikt kan worden voor asynchrone datatransmissie. Hij is geschikt voor datasnelheden van 300 bit/s (V.21) full duplex, 600/1200 bit/s (V.23) half duplex over tweedraadsverbindingen, en 1200/75, 75/1200 bit/s (Viewdata).
De drie werkingprincipes zijn 'auto-connect', 'auto-answer' en huurlijngebruik.
De EC 123 is verkrijgbaar als kaartmodel (Eurokaart) en als tafelmodel en kost fl. 595,= excl. b.t.w.

Ingezonden brief

Cursus Computerboekhouden van de LOI.

Van het LOI kregen we de volgende brief;

In Commodore Info nummer 8 bespreekt Jan Bodzinga een brief van een 64 gebruiker die problemen heeft met boekhouden op de computer.
De briefschrijver vertelt dat hij een boekhoudpakket voor zijn micro heeft aangeschaft waar hij slecht mee uit de voeten kan. Het programma werkt naar zijn zeggen bevredigend, maar hij ondervindt er toch problemen mee omdat het moeilijk is dit programma goed toe te passen zonder boekhoudkundige kennis.
Jan Bodzinga doet de briefschrijver twee suggesties aan de hand die eigenlijk allebei weinig bevredigd zijn. Niet zo verwonderlijk dat hij zijn advies aan de briefschrijver afsluit met de opmerking dat er behoefte is aan een goede cursus computerboekhouden. Maar helaas, zoiets bestaat volgens hem niet.
De slotopmerking van Jan Bodzinga is niet in overeenstemming met de feiten, want de LOI is enige tijd geleden gestart met een cursus Computerboekhouden. Deze cursus is bestemd voor iedereen die problemen ondervindt met geautomatiseerde boekhouding.
In de cursus wordt de theorie van het boekhouden uit de doeken gedaan. De cursist leert boekhouden met een speciaal voor de LOI ontwikkeld programma. Dit programma is geschikt voor de Commodore 64.
In tegenstelling tot de meeste boekhoudpakketten is de aanschaf noodzakelijk. De cursus Computerboekhouden is een uitkomst voor iedereen die met behulp van een 64 zijn eigen boekhouding wil gaan voeren. De cursus is verder geschikt voor iedereen die een eigen bedrijf heeft, administratief medewerker(ster) is of het graag wil worden.

Meer informatie kunt u krijgen bij de LOI. Bel 071-451308 en vraag naar de heer C.Braamzeel.

Ach, op deze manier krijgt het LOI toch weer wat aandacht voor zijn cursus, dus graag gedaan. Het ging er ons om ook de software-leveranciers erop te wijzen, dat die behoefte bestaat en dat bv. Tutor software bij een programma heel goed kan werken.
Red.

C-64 simuleert de 68000

De bekende Motorola 68000 processor (zit o.a. in de Amiga en Atari 520 ST) kan nu m.b.v. een 68000 pakket voor 8-bitters op de C-64 gesimuleerd worden. Echt veel praktische voordelen biedt deze simulatie nu niet bepaald, maar zij is wel leerzaam voor hen die meer over deze hypermoderne CPU willen weten.
Het 8 bit-pakket bestaat uit een OPAL 68000 Assembler (versie 1.03), de RSU 68000 Runtime Simulator (versie 1.07), de HOT 68000 Debugger (versie 1.07), twee (duitse) handboeken en twee referentie-kaarten met de bevelen voor de assembler en de Debugger. De handboeken gaan al van een gedegen voorkennis uit en mikken meer op een 68000-cursus voor gevorderde programmeurs.

Beach Head II

Met twee spelers tegen elkaar een echt gevecht kunnen aangaan, dat is het meest aantrekkelijke van Beach Head II van ACCES software. Maar natuuurlijk is er altijd de computer om de rol van de dictator over te nemen, die dan de eerste speler moet bevechten. Met veel variaties en tafrelen, ook via wat voice- simulatie wordt het geheel vrij realistich. Goede driedimensionele graphics en acceptabele schietscenario's, de afwisseling maakt dit spel aantrekkelijk. Zowel als helicopterpiloot als in de rol van infanterist kan men zich uitleven in 4 deelspelen.

EPROM-programmer-bouwkit

De firma Rainer Bohm, Postfach 2109, 4850 Minden in W.Dld. biedt een EPROM-programmer bouwpakket voor DM 333, -. Het apparaat kan standaard de EPROM-typen 2716, 2732, 2764, 27128 en 27256 met een spanning van 21 of 25 Volt programmeren. Via de RS232-uitgang is vrijwel elke hobbycomputer aansluitbaar. . De programmeertijden zijn zeer kort. Het programmeren van een 2764 EPROM kost nu nog maar 30 seconden i. p. v. de doorsnee zeven minuten.

Superbase voor C-64

Data Becker gaat nu eenpoging doen het professionele Database pakket voor de C-64. "Superbase" te marketen. Maar pas op, dit betreft de versie 1, er komt een Superbase versie 2 aan, voor ongeveer 399 gulden. SuperBase 1 is natuurlijk ook een superpakket voor gegevensbeheer, het biedt ondermeer: Een maximale datarecord van 1108 tekens; Een alleen door de floppy beperkte opslagcapaciteit; Een maximum van 127 velden;Gebruiksvriendelijke Reportgenerator;Data in/export; En een ingebouwde calculator. De database zelf kost slechts DM 198,-.
Het uitgebreide handboek DM 49, -.

Gebruikersgroepen

De **VCGN** (070-971851) is de grootste Commodore gebruikersvereninging en zit iedere laatste zaterdag van de maand in het Lodewijk Makeblijde lyceum in Rijswijk.
Inlichtingen over de vele afdelingen kunt u krijgen via:
Zeeland 01180-25746 West-Brabant 01640-51639 Midden-Brabant 01620-55634 Oost-Brabant 04120-42054 Zuid-Limburg 043-476935 Zuid-Holland 010-501023
De **HCC** Commodore gebruikersgroep heet Compet, 03404-59599.
De Commodore-club in Arnhem organiseert iedere twee weken op dinsdag clubavonden. Men geeft ook cursussen. Inl. Hofsingel 259, 6834 GH Velp, tel. 085-647782.
CCCB is een club in België met afdelingen in Antwerpen, Limburg, West-Vlaanderen. Inl. Bindstr. 19, 2600 Berchem Antwerpen. Lid worden door storting 500 fr op 405-5092171-90 tnv. CCCBA.
Ook in Gent is er een groep, te bereiken via 091-253937
HOT NEWS van SCN uit Amsterdam 020-104483 (alleen di. en wo. van 18.00-21.00 uur) heeft haar grote bijeenkomst in de Meervaart op de 2e zaterdag van de maand en ook activiteiten in Zwolle (038-548459) en Deventer (05700-50758).
PET Benelux heeft bijeenkomsten in Zeist (1e zaterdag), Nijmegen (2e zaterdag) en Haaksbergen (3e zaterdag). (Inl.05759-2211).
In **Noordwijk** zit een C-64/128 gebruikersgroep, die iedere 1e donderdag een bijeenkomst heeft. Inlichtingen A. v.d. Klaauw ☏ 01719-16720.
De bijeenkomsten van de gebruikersclub in het Belgische **Knokke-Heist** staan open voor gebruikers van alle merken home computers. Elke vrijdag vanaf 20.00 uur komen de computerenthousiasten uit de omgeving bijeen in Café Montyu in Knokke. Op zaterdag komt men bij elkaar in de Broederschool in Duinbergen van 9.30 tot 12.00 uur. Het contactadres is: Computerclub Knokke Heist, p.a. Bob Dyserink, Kerkstraat 19, 8390 Knokke Heist, België.

De HCC-dagen waren voor de de Nederlandse computer- enthousiaten een belangrijke gebeurtenis, maar het wereldnieuws werd in de afgelopen maand toch wel (of juist niet, zoals bleek) gemaakt in Las Vegas, waar de Comdex tentoonstelling nu s'werelds grootste computershow is. Luc Sala met een direct verslag.

Heet van de naald uit Las Vegas

COMDEX: 8 bit wereld ter ziele

In Amerika telt de 8 bit computer duidelijk niet meer mee, op een belangrijke tentoonstelling als de Comdex in Las Vegas wordt dat schrijnend duidelijk. Geen software, geen hardware, geen belangstelling van de fabrikanten voor wat men daar als een afgedane generatie computers ziet. Alles is 16 of 32 bit, wat de klok slaat.

Op het laatste moment liet Commodore zelfs verstek gaan op de Comdex; de gereserveerde ruimte liet men voor wat het was, en er kon niet meer af dan een wat schamele persconferentie in één van de hotels in deze stad vol casino's en speelautomaten. Daar had men een aantal dealers laten opdraven, die de aanwezige journalisten ervan moesten overtuigen, dat ze allemaal enorme zaken deden met de Amiga. Welaan, ik ben daar niet zo van overtuigd, en wel om twee redenen. Ten eerste is er een nieuwe versie van het operating systeem voor de Amiga in de maak, en dan ben je als computerverkoper toch wat voorzichtiger. Ten tweede heb ik maar een paar Amiga's (temidden van duizenden PC's en PC-compatibles) kunnen ontdekken. De aanwezige Amiga's stonden op de stands van Apple en Atari. In beide gevallen draaide men er demonstratieprogramma's op, met daarnaast hetzelfde of een soortgelijk programma op hun eigen machines, om te laten zien dat resp. de Mac en de 520 ST niet onderdeden wat betreft grafische prestaties. Nu is zoiets altijd wat bedrieglijk, men laat nooit iets zien, wat het tegendeel zou kunnen bewijzen.

Maar als de Amiga het grote succes is, waar Commodore op hoopt om uit de moeilijkheden te komen, dan zouden er toch wel een paar van de vele softwarehuizen op zo'n grote show als de Comdex, met meer dan 1200 exposanten, iets van hun kunnen op Amigagebied laten zien. Tot zover het hoofdstuk Amiga, ik denk, dat we die machine hier niet snel zullen zien. Zeker niet, voordat men de beloofde MSDOS aanpassing ook werkelijk kan leveren, en daarmee de levenskansen van de Amiga aanmerkelijk vergroot.



Halley's komeet, een bedreiging voor de homecomputer...?

Geen 64 ontwikkelingen

Wat geldt voor de Amiga op de show, gaat ook op voor de 64 in de computershops en supermarkten in de VS. Daar zijn de huiscomputers (óók die van Atari) vrijwel niet meer te vinden, hoogstens heeft men er nog een 128 staan, waarvan de verkoop wel redelijk loopt. De 64 is in de uitverkoop, net als de SX64, die men daar compleet met software en printer voor ongeveer 1500 gulden aanbiedt. Voor de 128 zijn in de VS vreemd genoeg wel 1571 diskdrives te koop met dubbelzijdig te beschrijven schijven, maar niet de bij ons nu gangbare 1570. Men spreekt wel over een dubbele diskdrive, die 1572 heet en binnenkort leverbaar wordt. Maar erg serieus nemen de winkeliers de computer voor de low-ends niet meer, dat is duidelijk. Maureen Fleming, onze correspondent in de VS, merkte op, dat de komst van de Halley komeet het einde van de huiscomputerrage markeert. Op dit moment is er namelijk een ware hausse in de gang op teleskoop gebied en waar vroeger de computers in bijv. de K-mart stonden, kan men nu kiezen uit tientallen verschillende modellen telescopen.

Geen ontwikkelingen

Op software-gebied was ik natuurlijk benieuwd naar de verwachte golf zakelijke software voor de 128. Maar ook daar enige teleurstelling. Als men al bereid is, om bepaalde software om de zetten naar het iets afwijkende CP/M formaat van de 128, dan gaat het om erg verouderde software. Bij Micropro bijvoorbeeld zei het hoofd van de ontwikkelingsafdeling, dat men niet meer werkt aan verbeteringen of aanpassingen van bv. Wordstar voor 8 bits computers. Die markt ziet men niet zitten, en dat in Europa nu toch honderdduizenden computerkopers voor het eerst kennis maken met CP/M voor hun 128, hun Schneider/Amstrad of hun MSX-2, interesseert ze daar in Amerika nauwelijks. Ook bij Ashton-Tate had men eigenlijk geen plannen, om dBase2 nog aan te gaan passen of te

verbeteren. Men beschouwt daar de 8 bit computers als een afgesloten boek. Jammer, het zijn juist die programma's, die het serieuze gebruik van met name de 128 zouden kunnen stimuleren. Men zou ze alleen wat moeten aanpassen aan de software- ontwikkelingen, zoals die bij de MS-DOS machine nu naar voren komen.

Software-trends

En bij die MS-DOS software-ontwikkelingen zitten er een paar, die ook voor ons de moeite waard zijn. De meest opvallende vernieuwingen vinden nu plaats op het terrein van de **instant interrupts**. Dat wil zeggen, dat men door het intikken van een bepaalde funktietoets of toetscombinatie vanuit het lopende programma direct een aantal hulpfunkties kan activeren. In deze categorie vallen bijvoorbeeld beeldschermcalculators, agenda's, en notitiebloks op het scherm. Dat is niet alleen voor de PC- achtigen mogelijk, Cardco liet bijvoorbeeld iets dergelijks in prototypevorm zien voor de 64. Met een druk op de knop krijgt men vanuit willekeurig welk programma een keuze-menu in beeld. Daarbij kan dan gekozen worden uit een rekenfunktie, notitieblokfunctie etc. Handig, al lopen we in ons land bepaald niet achter op dit gebied, sommige insteekmodules (bv. Power van KCS) hebben ook een dergelijke opzet, maar dan meer met utilities dan met toepassingsprogramma's.

Een volgende stap met instant interrupt programma's is die naar de direkte spellingscontrole. Voor de PC's is dat nu mogelijk, en we hopen, dat het ook voor de 128 en de 64 een haalbare zaak blijkt. Bij direkte spellingscontrole wordt al bij het intypen direkt door de computer gecontroleerd, of het betreffende woord wel goed wordt ingetypt. Is daar twijfel over (het woord staat niet in de woordenlijst), dan komt onmiddelijk het meest in de buurt komende woord in beeld. Dat is dan in 95% van de gevallen het correcte woord.

If you own an Apple IIc, you'd have to add all this

to match the versatility, expandability and higher intelligence of the new Commodore 128 (and it costs less too).

COMMODORE 128™ PERSONAL COMPUTER
A Higher Intelligence

In de U.S. wordt toch agressief voor de C-128 geadverteerd, zoals hier in Life Magazine

Wil men toch nog iets anders, dan is één keer een funktietoets indrukken al voldoende om een alternatief woord op te roepen. Dit is vooral bij het maken van tekst in vreemde talen een ware uitkomst (en engels is voor ons tenslotte een vreemde taal), en eigenlijk voor Europeanen van veel groter belang dan voor de mensen, die al van huis uit engels spreken. Of deze programma's ook al snel voor bv. nederlands en andere talen beschikbaar komen, werd niet duidelijk. De firma Borland liet bv. wel weten bezig te zijn met licentiebesprekingen over de Van Dale woordenlijsten, maar men denkt naar mijn idee daar in de VS wel iets te gemakkeijk over de nodige aanpassingen. In onze en andere europese talen spelen geslacht en naamval van een bepaald woord toch nog wel een grotere rol dan in het amerikaans.
Maar met dit soort programma's wordt toch duidelijk wel een stap gezet naar meer geperfectioneerde teksverwerking.
Luc Sala. ●

Voor wie is nieuwe Commodore bedoeld?
De diepten van de C-128

Of de ruim twee maanden geleden door Commodore gelanceerde C-128 werkelijk de successen zal oogsten die ons waren voorgespiegeld blijft na twee maanden nog een open vraag. Er zitten veel voordelen aan de C-128 maar aan de andere kant lijkt het erop dat lang niet alle hooggespannen verwachtingen direkt door dit nieuwe wonder van techniek kunnen worden gerealiseerd.
In dit artikel gaat Jan Bodzinga wat dieper in op de kwaliteiten die de C-128 ons te bieden heeft en voor wie de machine bizonder geschikt of juist wat minder geschikt is.

Nu de Commodore-128 een paar maanden te koop is, zijn we in de gelegenheid een wat diepere beschouwing van de opvolger van de C-64 te kunnen geven. Op dit moment zijn er in Nederland zo'n 4.000 machines van dit type in gebruik. Het grootste deel van deze 128'ers zijn in handen van de meest fervente Commodore aanhangers, die bijna een jaar lang hebben zitten popelen om de hand te kunnen leggen op het nieuwste ontwerp uit de Commodore serie. Een relatief klein percentage van de verkochte machines is terecht gekomen bij aspirant-computeraars, die altijd al de aanschaf van een huiscomputer overwogen en nu hun kans schoon zagen om direkt de meest geavanceerde huiscomputer te bemachtigen. Kortom, een redelijk succesvolle start van het fenomeen 128.

Vergelijken

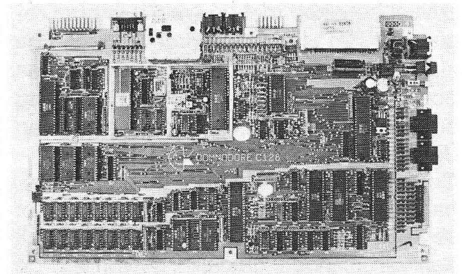
Wat de voordelen betreft, die de 128 te bieden heeft boven de bekende en beproefde C-64, lopen de meningen van de gebruikers nogal uiteen. We kunnen hierbij de nieuwelingen op het computergebied niet meetellen, omdat deze niet beschikken over de nodige ervaring ten opzichte van de capaciteiten van andere machines. Het is nu eenmaal moeilijk om te zeggen hoe goed de C- 128 voldoet aan de gestelde eisen, als je niet weet wat een ander type machine, zoals bv. een MSX of de nieuwe Atari 520 ST te bieden heeft.

Koopgedrag

Over het algemeen is het koopgedrag wat de C-128 betreft de eerste tijd nogal impulsief geweest. Men had gehoord van de geweldige mogelijkheden die door Commodore over de C-128 werden verspreid en op grond van deze informatie stroomden de intekeningen bij de dealers binnen voor een computer waarvan het nog maanden zou duren, voordat hij geleverd kon worden. Al weken van tevoren konden we in de catalogi van de diverse postorderbedrijven en warenhuizen genieten van een prachtige foto annex aanbieding voor deze gigant onder de huiscomputers.
Dat is de reden, dat de eerste dagen nadat de C-128 de magazijnen van Commodore verliet, de apparaten als warme broodjes over de toonbank gingen. Daarna is het wat stil geworden rond de C-128.
Dit is voor een deel te wijten aan de wat langzame levering van de machines door de importeur. Volgens de gegevens zijn er op 1 december zo'n 12.000 machines verkocht, waarvan er slechts 4.000 zijn geleverd. De andere 8.000 mensen wachten nog altijd op hun apparaat. Ze zijn dus niet aan te slepen en daarom spreekt de nederlandse importeur ook van een succesvolle start. En inderdaad, een verkoop van 6.000 machines per maand komt neer op meer dan 250 exemplaren per dag. Daarbij is wel de eerste hausse, direkt na het uitbrengen van de C-128, inbegrepen. Deze grootse aantallen zijn natuurlijk (nog) niet te vergelijken met de hoeveelheid C-64 machines die er in Nederland staan. De schattingen daarvan lopen uiteen van 150.000 tot 400.000!

64-Gebruik

Iedereen die enigszins op de hoogte is van de capaciteiten van de C-128 weet zo langzamerhand wel, dat nagenoeg **alle** software, die voor de C-64 voorhanden is, ook draait op de C-128. Daarbij moet wel worden aangemerkt, dat deze programma's daardoor op geen enkele manier beter, sneller of gemakkelijker te bedienen worden. Ook de bijpassende 1570 drive, die op zichzelf vele malen sneller is dan de oude 1541, brengt hierin geen verandering. Er blijken nogal wat C-64 bezitters te zijn, die denken dat de 1570- drive dezelfde overdrachtsnelheid kan uitvoeren met een C-64, waardoor ook de diskhandelingen op de 64 sneller zouden verlopen. Helaas is dit niet waar, want de C-64 beschikt niet over de nodige support om deze snelheid te kunnen verwerken. Het is wel mogelijk een 1570 aan te sluiten op de C-64, maar dan werkt hij even snel als de 1540/1541.
Ook op de C-128 in 64-mode is dit het geval. Daarnaast heb je bij de C-64 mode het nadeel, dat de specifieke toetsen, zoals het aparte numerieke toetsenbord en de vier extra cursortoetsen geen funktie hebben, zodat er ook in dat opzicht geen verschil bestaat met het werken op de C-64.
Gelukkig is mede door de hierboven aangemerkte 'nadelen' de C-128 bij het werken in de **64-mode** bijzonder compatibel met de echte C-64. Alle 64-software die we erop hebben geprobeerd, werkte perfect. En de ervaring van anderen is identiek. Het enige waarbij problemen kunnen ontstaan is het veranderde operating- systeem van de 1570/71 drives. Daardoor kan het voorkomen, dat bepaalde softwarebeveiligingen niet meer functioneren. Verder blijken er nogal wat cartridges te zijn, waarin bepaalde tool- commando's zitten die niet helemaal korrekt op de C-128 werken. Ook de C-64 reset (warme & koude start) verloopt anders in de C-128, omdat daar altijd wordt opgestart langs de processor en software van de 8510. Daardoor werken opdrachten als **unnew** en **old** niet altijd.
Al met al is Commodore bij het ontwerp van de C-128 er zondermeer in geslaagd een nieuwe computer op de markt te brengen, die volledig compatibel is met de C-



Het binnenwerk van de C-128

64. Dat betekent dat helaas ook alle minder aantrekkelijke kanten van de 64 in de C-128 zitten ingebouwd.

Het ontwerp

De makers van de 128 hebben als opdracht mee gekregen een computer te ontwikkelen, die naast de 6510 processor uit de 64 ook moest beschikken over een heel wat geavanceerder operatingsysteem met dito Basic-interpreter. Daaraan hebben we het geheel eigenzinnige en unieke ontwerp van de 128 te danken. Want waar op de wereld vind je een dergelijk stuk techniek? Eén computer die een simulatie geeft van drie types computer. Er zitten daarom ook 3 verschillende CPU's in het apparaat. De 6510 kennen we uit de 64. De nieuwe, soortgelijke 8510 processor heeft veel van de eigenschappen van de 6510 en de 6502 overgenomen, maar is daarnaast tweemaal sneller, terwijl er ook een paar extra instructies zitten ingebouwd. Een prachtig stuk werk, maar helaas niet erg gericht op de hedendaagse CPU-ontwikkelingen, die immers veeleer de Z80 serie supporten.
In feite is dit een trieste ontwikkeling. Zowel voor Commodore als voor de liefhebbers van een goed stuk machinetaal. Want welke assembly language is er nu fraaier dan die van de 6500-familie. Waar kun je zoveel bereiken met zo weinig interne registers? Het is als het vergelijken van de klanken van een prachtig stukje franse poëzie met het geluid van wat knauwerig amerikaans. Maar ja, het grootste deel van de wereld spreekt nu eenmaal engels.
Commodore heeft altijd al de neiging gehad bij de keuze van een standaard op het verkeerde paard te wedden. De oorzaak daarvan is nooit duidelijk geworden, de resultaten helaas des te beter.

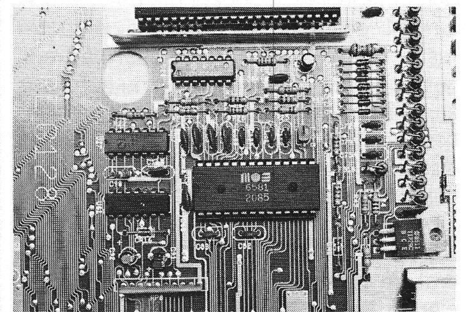
Misschien is dit wel de reden voor de **derde** chip die in de 128 zit. De Z80 microprocessor. Hierdoor kan deze computer werken met het bekende CP/M operating systeem van Digital Research, waarover we in het vorige nummer al uitgebreid berichtten.

Software

De software die voor de C-128 op de markt is, kunnen we in drieën verdelen. Ten eerste de bekende C-64 software, daarnaast de nieuwe C-128 programma's, waaronder diverse updates van beproefde C-64 applicaties. In dit gebied is er momenteel nog weinig te leveren, maar als de C-128 een commercieel succes gaat worden, zal daar wel verandering in komen. Nu kunnen we in feite alleen maar beschikken over **Superbase** en **Superscript** die voor de C-128 zijn aangepast en daardoor beter en sneller werken. Ook is Commodore Nederland bezig met het amerikaanse pakket **JANE** wat in grote lijnen alle applicaties voor een semi-zakelijke computertoepassing bevat, maar dit pakket is in Nederland nog niet te krijgen.
Als laatste bestaat er een bijzonder grote hoeveelheid CP/M software, die ooit ontwikkeld is voor de 8-bits voorloper van de PC. Deze software is tegen redelijke prijzen te koop, maar heeft een paar nadelen. Ten eerste worden deze programma's zonder uitzondering geleverd op een afwijkend diskformaat. Daardoor moet je tevens beschikken over een 1570/71 drive om er wat mee te kunnen doen. Bij het schrijven en lezen op afwijkende formaten heeft de C-128 het euvel nog langzamer te werken dan op zijn eigen formaat met CP/M 3.0. Het maken van kopieën naar Commodore formaat is mogelijk, maar wettelijk verboden, terwijl daarbij ofwel twee drives ofwel veel kennis van de CP/M plus het complete Operating Systeem komt kijken. Commodore zelf gaat geen CP/M software leveren op het eigen diskformaat. De beschikbare software is relatief erg verouderd, maar voldoet meestal nog prima voor de normale gebruiker. Er zijn per slot nog honderden bedrijven die werken met 8 bits machines.
De Basicinterpreter in C-128 mode, **Basic 7.0** is de meest uitgebreide Basic versie die op het moment te krijgen is. Het aantal commando's is meer dan verdubbeld ten opzichte van de C- 64. Over het nut van veel van deze opdrachten valt te twijfelen, zeker is wel, dat er veel minder vraag zal zijn naar toevoegingen in de vorm van cartridges, etc.

C-128 een succes ?

Zij, die de koop van een C-128 overwegen, moeten op een paar dingen letten. Daarbij is van doorslaggevend belang voor welke toepassingen de machine gebruikt moet worden.
De C-128 is in de eerste plaats nog altijd een **huiscomputer** waarmee spelletjes en vrij simpele data- en tekstbestanden kunnen worden verwerkt. In de zakelijke sfeer is het apparaat zeker sneller dan de C-64, maar kan zich nog altijd niet scharen onder



de grote groep van echte **Bussines-machines.** Ook de mogelijkheid van CP/M 3.0 draagt daar weinig aan bij.

Wil men dus een computer die in de eerste plaats voor professionele doeleinden moet worden gebruikt, dan is het niet verstandig de keuze op de C-128 te laten vallen. Daarentegen is deze machine ideaal voor de huis-, tuin- en keukenapplicaties en kan daarbij alle C-64 programmatuur verwerken, terwijl er voor de fervente Basic-programmeur een enorm scala van mogelijkheden is ingebouwd, waardoor de C-128 bijzonder geschikt wordt om eigen, snelle programma's voor uiteenlopende toepassingen te ontwikkelen.
De C-128 is enig in zijn soort, hét paradepaard van de homecomputers, maar ook niet meer dan dat.

Jan Bodzinga •

Champ/ Spritemachine

Nieuw pakket van Topsoft

Een paar weken geleden heeft het softwarehuis Topsoft uit Amsterdam een nieuw programmapakket het daglicht doen zien.

Op zichzelf niet zo bijzonder, ware het niet, dat het in dit geval een combinatie van twee unieke programma's betreft, zeker als de prijs die dit moois moet gaan kosten in de beoordeling wordt meegenomen.

Het nieuwe pakket is bedoeld voor de C-64 (en 128) en heeft als naam **Spritemachine ⋆ Champ** meegekregen. 'Spritemachine' staat voor de sprite-maker/controller terwijl 'Champ' het beste programma onder de machintetaal-assemblers heet te zijn. We zullen wat dieper in deze kampioen duiken om te zien in hoeverre de bewering van Topsoft op waarheid berust.

Champ-assembler

Om machinetaalprogramma's te kunnen schrijven, moet je als programmeur op z'n minst beschikken over een gezonde dosis kennis en over een hulpmiddel om de programma's te kunnen maken en te runnen. Dit hulpmiddel kan gevonden worden in 'Champ'. Dit programma bestaat uit vier onderdelen, een assembler, een editor, een debugger en een aparte monitor. De editor kan worden gebruikt om de assembleercode in mnemonics te schrijven. De debugger wordt daarna aangeroepen om de inmiddels via de assemblerfunctie tot machinecode vertaalde software op fouten te kunnen testen, terwijl de monitor de gebruiker in de gelegenheid stelt het RAM- en ROM-geheugen van de C-64 te kunnen bekijken.

Gebruik

Champ is voorzien van een uitgebreide nederlandse handleiding, waardoor het werken met dit programma in feite binnen ieders bereik ligt. Het programma is in eerste instantie gericht op het werken met een cassetterecorder. De opzet van het programma is erg gebruiksvriendelijk, zodat ook de beginnende machinetaal-programmeur zonder moeite aan de slag kan. Je kunt bijna zeggen, dat het programmeren niet fout kan gaan, omdat bij het intypen van iedere regel 'source-code' - in de editor - onmiddelijk wordt getest of er niet een foute opdracht is ingetypt. Dit maakt het programmeren een stuk aangenamer, zeker omdat er nogal eens een onnauwkeurige opdracht in de code kan voorkomen. De schermlay-out is overzichtelijk. Voor de drie tabellen: Labels, Opcodes en Adressen worden verschillende schermgradaties c.q. kleuren gebruikt. Het springen van de ene naar de andere tabel geschiedt eenvoudigweg met de cursor- en spatie-toetsen. Om een 'source-code' te assembleren kan worden volstaan met een 1-toets commando, waarbij een cijfer de gewenste opties aangeeft.

Deze opties lopen uiteen van een proef tot het uitgebreid listen op de printer, inclusief alle labels en adressen, van de complete code. Een complete lijst van mogelijke foutmeldingen is terug te vinden in de handleiding.

Toepassingen

Champ kan worden gebruikt met de C-64 en de C-128. Wat de 128 betreft, werkt Champ uitsluitend in de 64-stand, maar daarmee is het wel mogelijk om programma's te schrijven èn te assembleren die runnen in 128-stand. Bij het programmeren is een datasette noodzakelijk. Champ kan worden gebruikt voor alle soorten

INZENDVOORWAARDEN

Programma's en artikelen kunnen aan de redactie van Commodore-Info, PB 112, 1260 AC te Blaricum met vermelding REDACTIE CI, worden ingezonden en worden eventueel geplaatst, waarbij de inzender een door de redactie te bepalen vergoeding ontvangt, die ongeveer tussen de 50 en 50 gulden per pagina listings en de 100 en 200 gulden per pagina tekst ligt. Vermeld altijd naam, adres en vooral telefoonnummer. Alleen Listings, waarvan een cassette of diskette beschikbaar is, komen in aanmerking.
Gezien het relatief grote aantal inzendingen kan verwerking enige tijd op zich laten wachten. Indien we zeker niet tot plaatsing overgaan, ontvangt men de inzending retour, eventueel houden we het nog enige tijd als reserve.
SAC behoudt zich het recht voor, er redactionele wijzigingen in aan te brengen. Door inzending accepteert men, dat bij plaatsing alle publikatierechten, ook in electronische vorm, op de uitgever van het blad overgaan en dat men het blad vrijwaart van aanspraken van derden, bijvoorbeeld omdat het programma eigenlijk van een ander is of bv. deels of geheel uit een ander blad is overgenomen. Het overnemen van listings door lezers is toegestaan, maar verkoop van die listings, in welke vorm ook, aan derden is voorbehouden aan SAC.

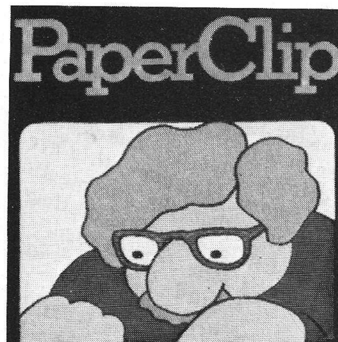
★SOFTWARE

machinetaalprogramma's waarbij alleen het beschikbare RAM-geheugen een limiet stelt aan de grootte.

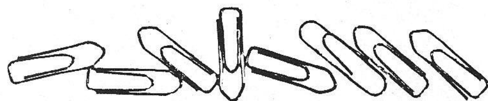
Waar voor je geld

Het gaat te ver om in dit concept uitvoerig alle functies te beschrijven die Champ te bieden heeft. De snelheid waarmee Champ ook erg grote machinetaalprogramma's kan assembleren ligt boven het gemiddelde. Het pakket is erop gericht ook de beginners stap voor stap in te wijden in het gebruik van het programma, al blijft een redelijke kennis van de 6502 machinetaal noodzakelijk. In de handleiding is een lijstje opgenomen met boeken etc. die hierover te koop zijn.
Ook het koppelen van Basic met machinetaal wordt behandeld.
Het grote voordeel van 'Champ' is de eenvoudige bediening waardoor de toch vrij ingewikkelde bezigheid van het programmeren in machinetaal bijna kinderwerk wordt.
Het pakket is te verkrijgen bij de betere computerhandel of rechtstreeks bij **Topsoft, Postbus 5220, 1007 AE Amsterdam.**
De prijs van de programmacombinatie Champ/Sprite-machine op cassette is f.35.-. ●

Een veelzijdige tekstverwerker zoekt iedereen. Easy Script is ongetwijfeld de bekenste tekstverwerker voor de C64 (en C128). Het mag dan wel de bekendste zijn, het programma heeft toch zo zijn minder prettige kanten. Een kijkje naar een andere word-processor kan daarom geen kwaad. Bijvoorbeeld Paperclip, een veelzijdige 80-kolomstekstverwerker voor de C64 en C128.



Paperclip



Vele C64-bezitters ergeren zich aan het slechts veertig tekens brede Commodore Scherm bij het tekstverwerken. Software als Easy Script mag dan wel over fraaie scroll- en video-outputfuncties beschikken. 40 Tekens op een regel blijft lastig werken. Een aantal fabrikanten (Zero 80,Video-pak-80) biedt 80- kolomskaarten. Helaas zijn deze stevig aan de prijs. Paperclip is zowel in een 40- als 80-teken versie leverbaar voor een aantrekkelijk bedrag.

Verder zijn er nog wat angelsaksische produkten die een 60-koloms display geven zoals Tasword. De bekendste tekstverwerkers voor de C64 buiten Easy Script (en opvolger Superscript) zijn Vizawrite, Textomat (Data Becker) en TV-tekst (Vosware). Deze pakketten zijn voor de hobbyist ontworpen en voldoen, met de dienovereenkomstige eisen in het achterhoofd, redelijk tot goed. Vizawrite is heel veelzijdig en ook wel populair bij de gebruikers ervan. Textomat biedt veel mogelijkheden, maar werkt tgv de vele menustappen tamelijk moeizaam. De prijs bedraagt c.a. *f* 100,-. TV-Tekst is een aardige word processor met diverse uitbreidingsmogelijkheden. Met name de snellaadoptie in kombinatie met de KCS-modules is aantrekkelijk. TV-tekst kost in de cassette uitvoering *f* 85,-. Paperclip heeft echter meer, vooral zakelijke noten op haar zang dan de bovenstaande tekstpakketten. En zij zou op menig kantoor niet misstaan.

Paperclip in het kort

Paperclip is een product van Batteries Included,Toronto. Het programma komt op een copieer-beveiligde disk en draait in de verschillende versies op de CBM 3000-,4000-,8000 zakenmachines en de C64/128 (in de 64-mode).
Paperclip is een niet-beeldscherm georiënteerde tekstverwerker. Dat wil zeggen dat de tekst ongeformatteerd in het geheugen is opgeslagen. Pas tijdens het printen krijgt de tekst zijn uiteindelijke vorm. Je zult de formatteringsinstructies dus zelf in de tekst moeten opgeven. Daardoor toont de schermtekst enigzins onoverzichtelijk. Het voordeel is echter dat de editor operaties vrij eenvoudig kunnen verlopen. Bovendien kan je de tekst in zijn toekomstige vorm via de video-output beoordelen. Paperclip kan zich meten met dure zakenpakketten en dat is niet gek voor een tekstverwerker van rond de *f* 200,-.

Unieke eigenschappen

Dat Paperclip meer is dan een doorsnee hobby word processor bewijzen een aantal van haar toch wel uniek te noemen eigenschappen:
-Behalve tekstblokken is het tevens mogelijk om kolommen te manipuleren. Daarmee kan je prachtige (kolommen-) tabellen maken.
-Een bijzonder krachtige replacement-functie.
- Standaardzinnen (One-Key Macros);Veel gebruikte zinnen, woorden of frasen kunnen ingeprogrammeerd en weer opgeroepen worden.
-Ingebouwde (engelse) spellingscorrector met vrije ruimte voor meer dan 5000 zelf te definieren woorden.
- Voorcontrole (Preview Mode); Bij vele tekstverwerkers geeft het 40 tekens tellende beeldscherm beoordelings- en correctieproblemen. In de Preview Mode van Paperclip is het mogelijk om horizontaal- en verticaal te scrollen en daarbij exact te kunnen zien of de woorden nu wel of niet op de pagina passen. In de normale mode geeft een diamandvormig sterretje het einde van de regel aan.
-Een 80-kolomsversie is leverbaar.
-Hoge gebruiksvriendelijkheid; Geen wisselende menu's meer,Alle opdrachten kunnen met enkele toetsindrukken op het "werkscherm" gegeven worden. Bovendien beschikt Paperclip over automatic save,letters/woord transposing en een optie om hoofd- in kleine letters e. o. over tezetten.
-Programeerbare pauze bij het afdrukken.
-Een naar believen instelbare regelbreedte.
-Een ingebouwde calculator.
-Multiple Disc File Global;Deze hele mondvol betekent gewoon dat Paperclip maarliefst 6 simultane search- en replace handelingen over een maximum van 4 disk drives kan uitvoeren.
-De mogelijkheid om ook sequential Files van andere tekstverwerkers te kunnen gebruiken.
-Twee koloms-afdrukmode.
-Werkt met meer dan 30 verschillende printers.
Verder kan de Paperclip-word processor natuurlijk alles wat de meeste andere tekstverwerkers ook presteren. Bijvoorbeeld onderstrepen, Super/Subscript, schuinschrift, diverse correctieprocedures, het plaatsen van kop/voetregels tabuleren, dupliceren, centreren, formatteren, printeraanpassingen enz.

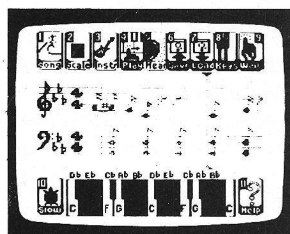
Toch niet helemaal ideaal?

Binnen de beperkingen van de C64 is Paperclip een vrijwel ideale tekstverwerker. Toch blijven er natuurlijk altijd wensen over. Bijvoorbeeld een grotere geheugenruimte of een betere 80- koloms display. Ook de formatterings opdrachten in de tekst komen soms storend over. De totaal indruk blijft echter een kwalitatief goede en veelzijdige tekstverwerker voor zo'n fl 200,- (40-kolomsuitvoering). Of het programma zich met de al aangekondigde nieuwe Commodore word processors kan meten zal de toekomst leren.

De Commodore heeft een erg uitgebreid scala aan geluidsmogelijkheden en dat is voornamelijk te danken aan de SID chip, op zichzelf al bijna een echte synthesizer. Maar ermee omgaan is niet eenvoudig, daarom dit artikel van F. Inklaar over geluidsmanipulatie met een programma om een en ander in de praktijk te kunnen toetsen.

SID : Muzikale capriolen





Wat is geluid eigenlijk en hoe komen we van electrische pulsjes tot hoorbaar geluid? Als een voorwerp staat te trillen, bijvoorbeeld een snaar of de conus van een luidspreker, brengt deze de omringende lucht ook aan het trillen, deze trillingen planten zich voort zoals watergolven bij een bewegende dobber, en zo ontstaan er golven in de lucht, die door onze oren waargenomen kunnen worden. Dit noemen we dan geluid, en de golven heten geluidsgolven.

Zo'n golf heeft nog een aantal belangrijke eigenschappen die ons oor waar kan nemen. Ten eerste de amplitude of golfhoogte, die we waarnemen als het volume van het geluid. Zo klinken lage golven zacht en hoge golven hard. Ten tweede is ons oor gevoelig voor de afstand tussen de golven, de zogenaamde golflengte van het geluid. Van golven met een korte golflengte zeggen we dat de toon hoog is, en van lange golflengte dat de toon laag is. Ten slotte kan ons oor golven van verschillende vormen onderscheiden. Zo kunnen we bijvoorbeeld het geluid van een trompet en een blokfluit uit elkaar houden, zelfs als op beiden dezelfde toon even hard wordt aangeblazen. Dit komt doordat twee instrumenten die op verschillende manieren de lucht aan het trillen zetten verschillend gevormde geluidsgolven doen ontstaan.

De omhullende/ envelope

Het zal duidelijk zijn dat tijdens het spelen van een toon op bijvoorbeeld een piano, het volume niet constant is. Zo wordt bij een piano de toon langzaam zachter, als we een toets aanslaan en ingedrukt houden. Als we het spelen op een piano dan na willen bootsen moeten we ervoor zorgen, dat de geproduceerde toon hetzelfde volumeverloop heeft. Het volumeverloop van een enkele gespeelde toon noemen we de omhullende, deze is karakteristiek voor het instrument en een belangrijke schakel in de geluidssynthese.

Wanneer we een instrument willen nabootsen of een speciale klank willen gaan maken is het aan te raden om eerst een schatting van de omhullende te maken, omdat de klank anders moeilijk te herkennen is. Op de SID en bij de meeste andere synthesizers leggen we de omhullende vast met 4 getallen. Deze heten de Attack, de Decay, de Sustain en de Release, normaal afgekort tot de **ADSR**. Met deze 4 getallen kunnen de omhullenden van vrijwel alle instrumenten worden nagebootst. Om aan te geven wat de ADSR inhoudt volgt hier een beschrijving van wat er tijdens het spelen van een noot met de omhullende gebeurt.

❶ We starten de toon. Dit is het moment waarop normaal een toets of een snaar of iets dergelijks wordt aangeslagen. Het volume stijgt nu snel (bv in 0.01 sec.) tot een maximum waarde. De tijd waarin dat gebeurt heet de **ATTACK-TIJD** of gewoon ATTACK.

❷ Hierna neemt het volume (meestal) iets langzamer (bv in 0.5 sec) af tot een constante waarde die het volume houdt zolang als we de toets ingedrukt houden of ergens op blijven blazen. De tijd waarin de afname plaatsvindt heet de **DECAY-TIJD** of DECAY, en het constante volume niveau het **SUSTAIN-LEVEL** of SUSTAIN. Let er vooral op dat de sustain een geluidsvolume is en geen tijdsduur!

❸ Als tot slot de toets of iets vergelijkbaars wordt losgelaten sterft de toon volledig uit. Het volume gaat dan naar nul in een tijdsduur die de **RELEASE-TIJD** of RELEASE heet.

In de handleiding van Commodore staat beschreven welke waarden er ingevoerd moeten worden om een bepaalde tijd voor de attack, decay of sustain te krijgen. Er zitten natuurlijk wel wat haken en ogen aan dit verhaal, zoals mag blijken uit de volgende voorbeelden.

1. Orgel

De toon zwelt nogal langzaam aan (A=8) en blijft maximaal zolang de toets ingedrukt blijft (S=15). Als de toets wordt losgelaten sterft de toon ook weer langzaam uit (R=10). Doordat de sustain maximaal is wordt de decay-fase overgesla-

gen. Die heeft dan ook geen invloed op het verloop en mag elke waarde hebben.

2. Piano

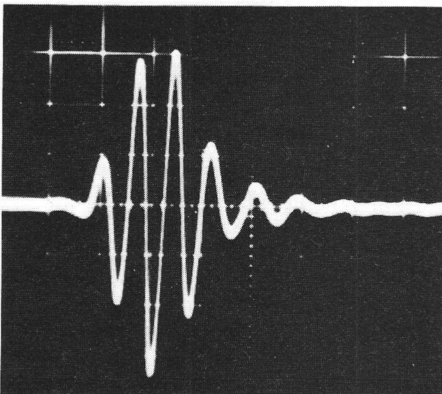
De toon is bijna meteen op volle sterkte (A=1). Als we de toets ingedrukt houden sterft de toon langzaam uit, het is niet mogelijk de toon aan te houden. Dit betekent dat de sustain op nul moet worden gezet (D=10, S=0). Als we de toets echter loslaten sterft de toon sneller uit (R=8). Bij de piano kunnen we dus direct van de decay- in de releasefase komen en wordt de sustainfase overgeslagen.

3. Xylofoon

Net als bij de piano is de toon snel maximaal en sterft langzaam uit. We kunnen hier de toon niet aanhouden, maar ook geen toets ingedrukt houden, zodat na de attackfase meteen de releasefase begint. Om praktische redenen zetten we dan de decay gelijk aan de release, omdat we dan de toon niet zo vlug uit hoeven te zetten. (Ga eens na dat dit hetzelfde effect heeft). (A=1, D=9, S=0, R=9) Het is dus duidelijk dat lang niet alle instrumenten alle vier de fasen werkelijk doorlopen. Ter geruststelling ook nog een instrument dat wel alle fasen doorloopt:

4. Trompet

(A=5, D=9, S=9, R=6). De attack en de decay vormen samen een piek bij het aanblazen, waarna de toon een constant volume aanneemt. Waneer we stoppen met blazen sterft de toon snel uit. Op dit punt is het verstandig om eens te gaan experimenteren met het programma. Let op de staaf met het bijschrift ENV(=envelope=omhullende). Kies eerst hoge waardes voor A,D en R zodat u met het oog kan volgen wat er gebeurt. Laat de toets los op verschillende momenten. Stel daarna kortere tijden in voor A,D en R en luister kritisch naar het resultaat.



GOLFVORMEN

In het eerste deel van dit artikel hebben we gezien dat de vorm van de geluidsgolf de klank van het geluid bepaalt. We zullen dit verband tussen vorm en klank nu eens onder de loep nemen. De meest bekende golfvorm is wel de zogenaamde sinusgolf, die eruit ziet als een ongestoorde golf op het water. Hij is onder meer te horen als Hilversum 3 er na twaalven mee op houdt. Deze golf klinkt wat kaal en is op de duur zelfs irritant, zodat je hem in de muziek vrijwel niet tegen zult komen, maar hij speelt een zeer belangrijke rol bij het begrijpen van andere golven.
Het blijkt namelijk dat alle mogelijke golfvormen gemaakt kunnen worden door een juiste combinatie van sinusgolven. We zullen dit beschouwen aan de hand van een snaar, maar het idee gaat voor alle muziekinstrumenten op. Het is handig als u voor dit stuk een gitaar bij de hand hebt, maar u kunt ook een stuk elastiek tussen twee punten spannen. Bij de gitaar moet u de (dikste) 6e snaar - dat is de lage E-snaar - gebruiken.
Als we de snaar nu in het midden aanslaan begint hij met een zekere frequentie te trillen, waarbij de snaar in het midden de grootste amplitude heeft, en het afneemt tot nul aan de zijkanten. Deze manier van

trillen heet de eerste normaaltrilling, en de toon (hier dus de lage E) heet de grondtoon of de eerste **harmonische.**

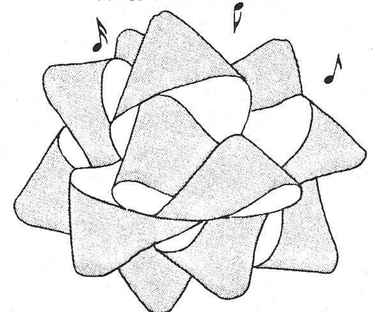
Als we nu voorzichtig een vinger precies halverwege de snaar (waar de hals en de kast samenkomen) op de snaar leggen, zonder de snaar in te drukken, en de snaar weer aanslaan, komt de snaar in de tweede normaaltrilling: De snaar staat stil aan de randen en in het midden, en de amplitude is maximaal op 1/4 en 3/4 van de snaar. De toon die de snaar nu produceert, heet de tweede harmonische en blijkt precies twee maal zo hoog te zijn als de grondtoon. In de muziek komt dat neer op precies een octaaf hoger (de middelste E). Met wat oefening lukt het ook nog wel de derde en vierde normaaltrillingen tevoorschijn te halen, door respectievelijk de snaar af te dempen boven het zevende en het vijfde fret. Op die manier genereren we de derde harmonische, die drie maal de frequentie van de grondtoon heeft (middelste B, octaaf + quint) en de vierde, die vier maal zo hoog is (hoge E, twee octaven). In theorie kan je hier oneindig lang mee door gaan, maar in de praktijk is dat niet zo gemakkelijk.

Als een snaar wordt aangeslagen voert hij al deze normaaltrillingen tegelijk uit, hoewel het op het oog lijkt dat alleen de eerste normaaltrilling optreedt. Dit komt doordat die veruit het sterkste is, en de verdere normaaltrillingen alleen maar een kleine verstoring van de snaarvorm veroorzaken. Elk van deze normaaltrillingen brengt een zuivere sinusgolf voort, maar het is onmogelijk om met de hand maar een normaaltrilling tegelijk aan te slaan. Toen we bijvoorbeeld de snaar in het midden aansloegen om de eerste normaaltriling te zien, sloegen we ongewild ook de derde, de vijfde, de zevende enz. aan. Daarom was de toon die we toen hoorden geen sinusklank. Als we nu, zoals tijdens normaal spel, de snaar op een willekeurige plaats aanslaan, worden dus alle normaaltrillingen aangeslagen en produceert de snaar een grondtoon, plus twee keer de grondtoon, drie keer, vier keer etc..

Het vreemde is nu dat we die tonen niet apart horen, maar dat de tonen met elkaar versmelten tot een soort akkoord, wat vaak een aanzienlijk prettiger klank heeft dan alleen een sinusgolf.

Normaaltrillingen De klank blijkt af te hangen van welke normaaltrillingen er onstaan, en hun onderlinge amplitude- of sterkteverhoudingen. We kunnen die variëren door de snaar op verschillende plaatsen aan te slaan. Precies in het midden krijgen we bijvoorbeeld alleen de oneven harmonischen, wat een beetje dof klinkt, en als we de snaar vlak aan de randen aanslaan worden de harmonischen relatief harder, waardoor de klank wat scheller wordt. Net als bij de klank blijken ook de geluidsgolven van de verschillende harmonischen niet afzonderlijk waarneembaar te zijn: de geluidsgolven vormen samen een nieuwe golf, die echter niet sinusvormig meer is.

Samenvattend kunnen we dus zeggen dat verschillende verhoudingen tussen de harmonischen aanleiding geven tot een verschillende klank en een verschillende golfvorm. Bij synthesizers maken we van het omgekeerde gebruik: Als we verschillende golfvormen opwekken, die verschillende verhoudingen van harmonischen hebben, zullen die ook anders klinken.



SYNTHESE

Stel nu dat we van een instrument de verhoudingen van de harmonischen kennen en dat we dat instrument elektronisch na willen bootsen. De eerste mogelijkheid is dan om een stuk of tien sinusgeneratoren te nemen, die af te stellen op de frequenties van de harmonischen en in de juiste verhoudingen bij elkaar te mengen. Deze methode heet **additieve synthese** en is niet erg populair vanwege het aantal sinusgeneratoren dat nodig is om 1 toon te produceren.

Een veel gebruikelijker methode is de **subtractieve synthese** die als volgt werkt: je genereert een of andere complexe golfvorm die van nature flink wat boventonen bezit, met een bekende verhouding. Normaal zal dit signaal dus teveel boventonen bevatten. Dit is elektrisch heel eenvoudig te verhelpen door het signaal door een af te regelen filter te sturen, dat bepaalde frequenties, dus ook bepaalde boventonen, beter doorlaat dan andere. Achter het filter zijn dan de boventoonverhoudingen veranderd, en het filter beinvloedt de klank op een regelbare manier.

Deze methode verdient meestal de voorkeur, alleen al doordat het elektrisch gemakkelijker te bouwen valt. Ook in de SID-chip wordt deze methode van subtractieve synthese gebruikt. Voor het ongefilterde basissignaal voor subtractieve synthese is het volgende viertal golfvormen populair:

1. De driehoeksgolf

Van alle beschikbare golfvormen lijkt deze het meest op een sinusgolf. Alleen de oneven harmonischen zijn aanwezig en hun sterkte neemt in volgorde van toenemende toonhoogte snel af. Daardoor is deze toon zacht van klank, en bijzonder geschikt voor houtblazers en houten slaginstrumenten.

2. De zaagtand.

Deze toon bevat alle boventonen en ze zijn veel sterker dan bij de driehoeksgolf. Hierdoor is de klank wat feller, wat hem vooral geschikt maakt voor snaarinstrumenten.

3. De blok/pulsgolf.

De pulsgolf is een golf met maar twee niveau's die elkaar snel afwisselen. Dit geeft het geluid een schelle klank, bekend uit de beginjaren van het computergeluid. Het percentage van de tijd dat het signaal hoog is noemen we de duty-cycle of pulsewidth(=puls breedte). Door hier geschikte waarden voor te kiezen kunnen we bepaalde boventonen weg laten vallen. Bij bijvoorbeeld een blokgolf die een duty cycle 50% heeft, d.w.z. dat het signaal net zo lang hoog als laag is, vallen alle even boventonen weg. Wat we dan overhouden is dus de grondtoon, 3 x de grondtoon, 5 x de grondtoon enzovoort. Blokgolven kunnen onder meer gebruikt worden voor koperblazers.

4. Ruis.

Ruis zal in het algemeen niet voor muziek gebruikt worden. Het is nuttig bij speciale effecten zoals explosies maar kan ook gebruikt worden om klanken te perfectioneren, bijvoorbeeld pijpruis bij een orgel of een fluit. Ruis bevat niet slechts bepaalde harmonischen maar bevat alle frequenties. Als daarbij de hoge frequenties sterker zijn spreken we van 'blue noise', bij lage frequenties van 'pink noise', en als alle frequentiegebieden even sterk zijn van 'white noise'. Bij de SID valt dit te regelen met de frequency-parameters, maar ook met het filter kan achteraf de ruisklank worden veranderd.

Filteren

Zoals al eerder werd vermeld, worden filters gebruikt om de harmonische inhoud van de basisgolfvormen te veranderen, om langs die weg andere klanken te maken. De SID chip bevat hiervoor alle benodigde filters, die naar believen aan of uit kunnen worden gezet en veranderd kunnen worden in frequentie.

1. **Lowpass- of laagdoorlaatfilter**

Zoals de naam al zegt heeft dit filter de eigenschap lage frequenties door te laten. Hoge frequenties worden door dit filter onderdrukt. De grens tussen deze twee gebieden kunnen we verschuiven door het bijstellen van de filter cut off frequency. Zo kunnen we dus het Lowpass filter zo instellen, dat de boventonen veel meer worden onderdrukt dan de grondtoon,die immers de laagste frequentie heeft. De klank zal daardoor in het algemeen wat zachter worden.

2. **Highpass- of hoogdoorlaatfilter**

Dit is dus de omgekeerde versie van het lowpass filter. Hiermee kunnen we de grondtoon enigszins onderdrukken, waardoor de toon wat scheller wordt.

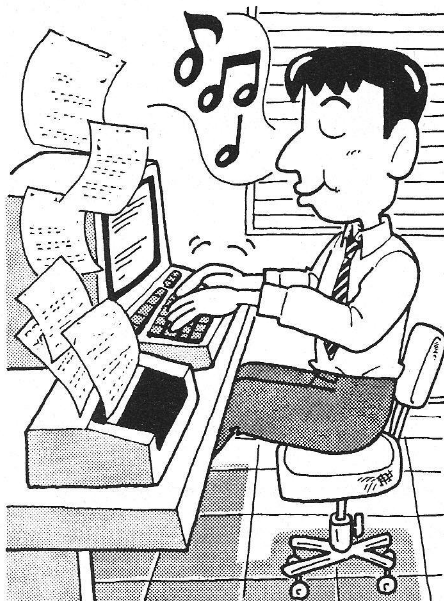
3. **Bandpass- of banddoorlaatfilter**

Dit filter onderdrukt zowel hoge als lage tonen, maar laat tonen in het 'middengebied' rond de cut-off frequency door. Met dit filter kunnen we een speciale boventoon meer naar voren halen, wat mooie klanken kan opleveren.

1+2 Notch reject filter

De SID staat ook toe meerdere filters tegelijk aan te schakelen, waarbij een frequentie wordt doorgelaten wanneer hij door minstens een van beide filters wordt doorgelaten (Additief filteren). De meest interessante mogelijkheid is daarbij de kombinatie van high- en lowpass. Zowel de hoge als de lage tonen worden dan doorgelaten, maar juist de tonen in het middengedeelte worden onderdrukt, zij het niet zo sterk als bij de andere filters. Dit schept dus de mogelijkheid specifieke boventonen te onderdrukken.

Meestal bieden filters ook nog de mogelijkheid om de resonantiefactor in te stellen. Dit effect versterkt frequenties in de buurt van de cut-off frequency en kan mooie effecten opleveren.

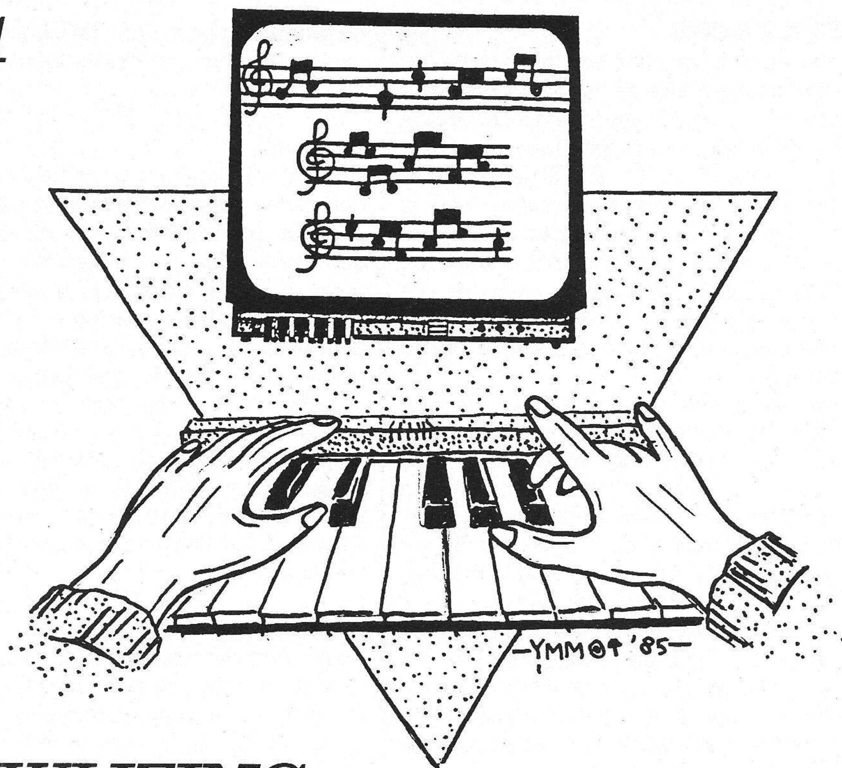


HET PROGRAMMA

Na al deze theorie is het zeker nodig om het een en ander eens in de praktijk uit te proberen, want het is onmogelijk om geluidssynthese alleen vanaf papier te leren. Het programma **soundmon** biedt de mogelijkheid om eenvoudig parameters te veranderen en het resultaat te beluisteren. Bovendien kan het verloop van de omhullende (envelope) met het oog worden gevolgd en is het mogelijk om van een geproduceerde klank de golfvorm en de harmonische inhoud te laten bepalen. Het programma kent echter ook een paar beperkingen: Bij de bepaling van de harmonische inhoud wordt geen rekening gehouden met het filter. Bovendien is het mogelijk om meerdere golfvormen tegelijk aan te zetten om op die manier andere golfvormen te genereren. Maar door een voor mij onbekende reden zijn de analyse resultaten dan niet betrouwbaar. Tenslotte meld ik nog even dat het vooral bij vroege versies van de C-64 kan voorkomen dat de SID niet helemaal doet wat hij zou moeten doen, zo kan de omhullende soms blijven hangen in plaats van uit te sterven voor een bepaalde release-waarde. Wees daar dus op verdacht.

Vragen liefst schriftelijk naar: F. Inklaar, Kemphaanstr. 3, 1531 VA Wormer of via VIDITEL nr. 400019556 (E.A.I. Automatisering.)

SOUNDMON 1.1



Een programma om geluidseffecten en muziek te maken

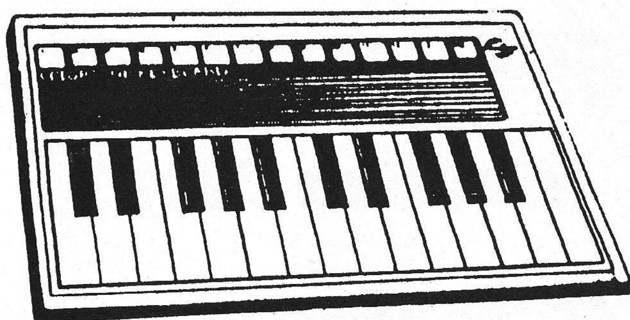
GEBRUIKSAANWIJZING

Het scherm bestaat uit vier delen:
Linksboven staat de **ADSR**-controle.
Rechtsboven kan de golfvorm worden ingesteld.
Rechtsonder staan alle parameters die betrekking hebben op het filter.
Het gedeelte linksonder geeft informatie aan de gebruiker over de golfvorm en de ADSR.
De eerste drie gedeeltes dienen om het geluid in te stellen. Daarvoor hoort bij elke parameter de toets, waarvan de letter in een afwijkende kleur op het scherm staat. Als die toets wordt ingedrukt wordt de waarde verhoogd of de golfvorm of filter ingeschakeld. In combinatie met de shift toets wordt de waarde verlaagd of het effect uitgeschakeld. Bij de **pulsewidth** is er bovendien nog een fijnregeling mogelijk met de + en de - toetsen.
Een eenvoudig muziekklavier vormen dan de toetsen 3 tot en met 0. Het gedeelte linksonder bevat links een kolom waarop continu de waarde van de omhullende wordt bijgehouden, om de invloed van de ADSR met het oog te kunnen volgen. De twee vakjes ernaast beelden respectievelijk de ingestelde golfvorm en de harmonische inhoud daarvan af.
Omdat het berekenen daarvan veel tijd kost wordt dit niet continu gedaan maar alleen na het indrukken van de RETURN-toets. De rand van het scherm wordt dan donker, om aan te geven dat het programma geen instructies kan verwerken en de vorm en inhoud worden getekend. Zodra dit is afgerond wordt de randkleur weer normaal. Tot slot nog twee zaken:

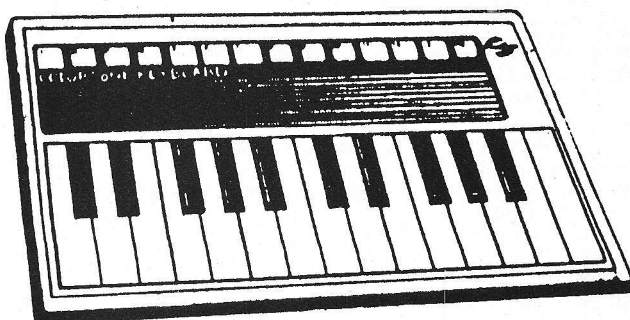
1. De berekende harmonische inhoud slaat op het ongefilterde signaal.

2. Er mogen meerdere golfvormen tegelijk worden aangezet. De analyse levert dan alleen geen zinnige resultaten op.

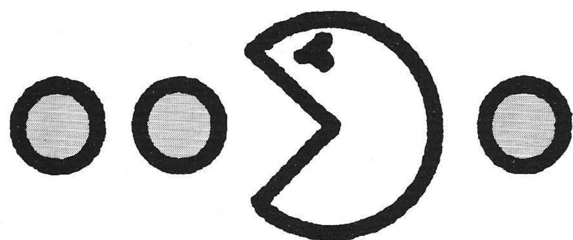
```
100 REM ***********SOUNDMON 1.1****************
110 REM *                                     *
120 REM * COPYRIGHT 1985 E.A.I AUTOMATISERING *
130 REM *                                     *
140 REM *       PROGRAM BY F. INKLAAR         *
150 REM ***************************************
160 SID=54272
170 PRINT"[illegible]                EVEN GEDULD A.U.B."
180 DIM A(128), NT(37), AT$(15), DE$(15), SU$(15),
    RE$(15), SI(360), CO(360)
190 FOR I=1 TO 37: READ NT(I): NEXT
200 FOR I=1 TO 8: READ LF(I): NEXT
210 FOR I=1 TO 8: READ HF(I): NEXT
220 I=0: J=0
230 READ A: IF A=-1 GOTO 250
240 POKE 49152+I,A: J=J+A: I=I+1: GOTO 230
250 READ A: IF A<>J THEN PRINT"FOUT IN DATA
    (REGEL 5100-5310)": STOP
260 SYS 49152: REM START INTERRUPT ROUTINE
270 FOR I=0 TO 24: POKE SID+I,0: NEXT:
    REM RESET SID
280 FOR I=0 TO 360
290 CO(I)=COS(I*π/180)
300 SI(I)=SIN(I*π/180)
310 NEXT
320 REM * SET SPRITES *
330 POKE 53248,72: POKE 53249,150
340 POKE 53250,136: POKE 53251,150
350 POKE 53271,3: POKE 53277,3
360 POKE 2040,13: POKE 2041,14
370 SP=0: GOSUB 3600
380 SP=1: GOSUB 3600
390 POKE 53287,14: POKE 53288,14: POKE 53280,14
400 POKE 53269,3
410 REM * INIT. VARIABELEN *
420 AT=3: DE=7: SU=5: RE=9
430 PW=2048: FS=5: VO=15: TR=1
440 POKE 650,128
450 REM INIT. STRINGS
460 AT$="[illegible]"
470 DE$="[illegible]"
480 SU$="[illegible]"
490 RE$="[illegible]"
500 FOR I=0 TO 15
510 READ A$, B$
520 AT$(I)=AT$+A$
```



```
530 DE$(I)=DE$+B$
540 SU$(I)=SU$+STR$(I)+"/15 "
550 RE$(I)=RE$+B$
560 NEXT I
570 A$="▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮"
580 TR$=AT$+A$
590 SA$=DE$+A$
600 PU$=SU$+A$
610 PW$=RE$+A$
620 A$="▮▮▮▮▮▮▮▮▮▮▮▮"
630 LP$=TR$+A$
640 BP$=SA$+A$
650 HP$=PU$+A$
660 RS$=PW$+A$
670 FR$=RS$+"▮▮"
680 AC$(1)=" ON"
690 AC$(0)="OFF"
700 GOSUB 4000: REM SCHERM SET UP
710 GOSUB 2300: REM UPDATE SCHERM+SID
1000 GETA$: IF A$<>"" THEN 1000: REM CLEAR BUFFER
1010 IF PEEK(197)=64 GOTO 1010
1020 A=PEEK(197): GET A$: IF A$="" GOTO 1020
1030 IF A>37 GOTO 1060
1040 NT=NT(A): IF NT=0 GOTO 1060
1050 GOTO 1500
1060 REM * COMMANDO *
1070 IF A$=CHR$(13) THEN GOSUB 3000: GOTO 1000:
     REM ANALYSEER
1080 GOSUB 2000: GOTO 1000: REM WIJZIG PARAMETERS
1500 REM * START TOON *
1510 FF=328*LOG(FS*(256*HF(NT)+LF(NT)))-1963
1520 FH=INT(FF/8)
1530 FL=FF-FH*8
1540 POKE SID+14,LF(NT): POKE SID+15,HF(NT)
1550 POKE SID+21,FL: POKE SID+22,FH
1560 POKE SID+18,WV+1: REM START TOON
1570 IF PEEK(197)=A GOTO 1570 :REM WACHT OP
     LOSLATEN TOETS
1580 POKE SID+18,WV: REM START RELEASE CYCLE
1590 GOTO 1000
2000 IF (A$="A") AND (AT<15) THEN AT=AT+1:
     GOTO 2300
2010 IF (A$="♠") AND (AT>0) THEN AT=AT-1: GOTO 2300
2020 IF (A$="D") AND (DE<15) THEN DE=DE+1:
     GOTO 2300
2030 IF (A$="‾") AND (DE>0) THEN DE=DE-1: GOTO 2300
2040 IF (A$="S") AND (SU<15) THEN SU=SU+1:
     GOTO 2300
2050 IF (A$="♥") AND (SU>0) THEN SU=SU-1: GOTO 2300
2060 IF (A$="R") AND (RE<15) THEN RE=RE+1:
     GOTO 2300
2070 IF (A$="_") AND (RE>0) THEN RE=RE-1: GOTO 2300
2080 IF A$="P" THEN PU=1: GOTO 2300
2090 IF A$="┐" THEN PU=0: GOTO 2300
2100 IF A$="O" THEN SA=1: GOTO 2300
2110 IF A$="┌" THEN SA=0: GOTO 2300
2120 IF A$="T" THEN TR=1: GOTO 2300
2130 IF A$="|" THEN TR=0: GOTO 2300
2140 IF (A$="+") AND (PW<4095) THEN PW=PW+1:
     GOTO 2300
2150 IF (A$="W") AND (PW<3996) THEN PW=PW+100:
     GOTO 2300
2160 IF (A$="-") AND (PW>0) THEN PW=PW-1: GOTO 2300
2170 IF (A$="o") AND (PW>100) THEN PW=PW-100:
     GOTO 2300
2180 IF A$="H" THEN HP=1: GOTO 2300
2190 IF A$="|" THEN HP=0: GOTO 2300
2200 IF A$="L" THEN LP=1: GOTO 2300
2210 IF A$="L" THEN LP=0: GOTO 2300
2220 IF A$="B" THEN BP=1: GOTO 2300
2230 IF A$="|" THEN BP=0: GOTO 2300
2240 IF (A$="F") AND (FS<10) THEN FS=FS+.25:
     GOTO 2300
2250 IF (A$="-") AND (FS>.25) THEN FS=FS-.25:
     GOTO 2300
2260 IF (A$="V") AND (VO<15) THEN VO=VO+1:
     GOTO 2300
2270 IF (A$="X") AND (VO>0) THEN VO=VO-1: GOTO 2300
2280 IF (A$="E") AND (RS<15) THEN RS=RS+1:
     GOTO 2300
2290 IF (A$="‾") AND (RS>0) THEN RS=RS-1: GOTO 2300
2300 FO=0:IF (HP=1) OR (LP=1) OR (BP=1) THEN FO=1
2310 WV=64*PU+32*SA+16*TR
2500 REM UPDATE SCHERM
2510 PRINT AT$(AT), DE$(DE), SU$(SU), RE$(RE)
2520 PRINT PU$ AC$(PU), TR$ AC$(TR), SA$ AC$(SA)
2530 A$=STR$(PW): PRINT PW$+A$+LEFT$("
     ",5-LEN(A$))
2540 REM (↑ VUL STRING AAN TOT LENGTE 4 MET
     SPATIES)
2550 PRINT LP$ AC$(LP), HP$ AC$(HP), BP$ AC$(BP)
2560 A$=STR$(RS): PRINT RS$+A$+LEFT$("
     ",3-LEN(A$))
2570 A$=STR$(FS): PRINT FR$+A$+LEFT$("
     ",5-LEN(A$))
2580 REM UPDATE SID
2590 I=INT(PW/256): J=PW-I*256
2600 POKE SID+16,J: POKE SID+17,I
2610 POKE SID+19,16*AT+DE
2620 POKE SID+20,16*SU+RE
2630 POKE SID+23,16*RS+4*FO
2640 POKE SID+24,64*HP+32*BP+16*LP+VO
2650 RETURN
3000 F=16: POKE 53287,6: POKE 53288,6:
     POKE 53280,2
3010 POKE SID+14,F: POKE SID+15,0
3020 POKE SID+18,WV+8: POKE SID+18,WV
3030 FORI= 1 TO 122: A(I)=PEEK(SID+27): NEXT I
3040 POKE 53287,14
3050 SP=0: GOSUB 3600: FOR X=1 TO 20
3060 YM=INT(A(INT(X*6.4))/14)
3070 FOR Y=1 TO YM: GOSUB 3500
3080 NEXT Y,X
3090 REM * FOURIER TRANSFORMATIE *
3100 POKE 53288,14
3110 SP=1: GOSUB 3600
3120 FOR N=1 TO 10
3130 LA=122/N: S=0: C=0
3140 FOR I=1 TO 122
3150 J=I/LA-INT(I/LA)
3160 S=S+A(I)*SI(J*360)
3170 C=C+A(I)*CO(J*360)
3180 NEXT I: IF (S=0) AND (C=0) THEN 3280
3190 AM(N)=3*LOG(S*S+C*C)/LOG(2)
3200 IF N=1 THEN AM(0)=AM(1)
3210 FOR Y=1 TO YM: GOSUB 3500
3220 X=2*N-1: YM=INT(19-(AM(0)-AM(N))/1.5)
3230 IF YM<0 THEN 3270
3240 IF YM>19 THEN YM=19
3250 FOR Y=1 TO YM: GOSUB 3500
3260 NEXT Y
3270 NEXT N
3280 POKE 53280,14: RETURN
3500 REM * PLOT IN SPRITE *
3510 XR=X+2: YR=22-Y
3520 BY=3*YR+INT(XR/8)
3530 BI=7-(XR-INT(XR/8)*8)
3540 BY=BY+832: IF SP=1 THEN BY=BY+64
3550 POKE BY,(PEEK(BY)OR(2↑BI))
3560 RETURN
3600 BY=832: IF SP=1 THEN BY=BY+64
3610 FOR I=0 TO 2: POKE BY+I,255: NEXT
3620 FOR I=5 TO 59 STEP 3: POKE BY+I,1: NEXT
3630 FOR I=62 TO 60 STEP -1: POKE BY+I,255: NEXT
3640 FOR I=57 TO 3 STEP -3: POKE BY+I,128: NEXT
3650 FOR I=4 TO 58 STEP 3: POKE BY+I,0: NEXT
3660 RETURN
4000 PRINT"▮┌──────────────────┬──────────────────┐";
4010 PRINT"|▮A▮TTACK   :        |▮T▮RIANGLE   :   /\|";
4020 PRINT"|                    |                  |";
4030 PRINT"|▮D▮ECAY    :        |SAWT▮O▮OTH   :  /|  |";
4040 PRINT"|                    |                  |";
```



```
4050 PRINT"|SUSTAIN :                |PULSE      :  ┌┘|";
4060 PRINT"|                         |              |";
4070 PRINT"|RELEASE :                |PULSEWIDTH:    |";
4080 PRINT"|                         |              |";
4090 PRINT"├────┬────────────────────┴┬─────────────┤";
4100 PRINT"|OMH|                      |              |";
4110 PRINT"|┌─┐|   VORM INHOUD        |LOWPASS   :   |";
4120 PRINT"|| ||                      |              |";
4130 PRINT"|| ||                      |BANDPASS  :   |";
4140 PRINT"|| ||                      |              |";
4150 PRINT"|| ||                      |HIGHPASS  :   |";
4160 PRINT"|| ||                      |              |";
4170 PRINT"|| ||                      |RESONANCE :   |";
4180 PRINT"|| ||                      |              |";
4190 PRINT"|| ||                      |FREQUENCY :   |";
4200 PRINT"|└─┘|                      |              |";
4210 PRINT"└────┴──────────────────────┴──────────────┘";
4220 PRINT"                    SOUNDMON 1.1"
4230 PRINT"           TERRESTRIAL SOFTWARE 1985";
4240 RETURN
5000 REM NT ARRAY (CURRENT KEY > NOOT)
5010 DATA 0,0,0,0,0,0,0,1,0,0
5020 DATA 2,0,0,0,0,3,0,0,4,0
5030 DATA 0,0,0,5,0,0,6,0,0,0
5040 DATA 0,7,0,0,8,0,0
5050 REM LF ARRAY (LOW FREQUENCIES)
5060 DATA 37,63,154,227,177,214,94,75
5070 REM HF ARRAY (HIGH FREQUENCIES)
5080 DATA 17,19,21,22,25,28,32,34
5090 REM MACHINETAAL INTERRUPT ROUTINE
5100 DATA 120, 169,  13, 141,  20,   3
5110 DATA 169, 192, 141,  21,   3,  88
5120 DATA  96, 234,  72, 152,  72, 138
5130 DATA  72, 160,   0, 162,   8, 169
5140 DATA 250, 133, 253, 169,   6, 133
5150 DATA 254, 173,  28, 212, 141, 108
5160 DATA 192, 224,   0, 240,  58, 202
5170 DATA 138,  72, 173, 108, 192, 201
5180 DATA  32, 176,  17, 106,  24, 106
5190 DATA 170, 189, 109, 192, 145, 253
5200 DATA 169,   0, 141, 108, 192,  76
5210 DATA  81, 192, 169, 160, 145, 253
5220 DATA  56, 173, 108, 192, 233,  32
5230 DATA 141, 108, 192,  56, 165, 253
5240 DATA 233,  40, 133, 253, 165, 254
5250 DATA 233,   0, 133, 254, 104, 170
5260 DATA  76,  37, 192, 234, 104, 170
5270 DATA 104, 168, 104,  76,  49, 234
5280 DATA   0,  32, 100, 111, 121,  98
5290 DATA 248, 247, 227, -1
5300 REM CHECKSUM
5310 DATA 15693
5320 REM STRINGS VOOR SCHERMOUTPUT
5330 DATA " 0 (   2 MS)"," 0 (   6 MS)"
5340 DATA " 1 (   8 MS)"," 1 (  24 MS)"
5350 DATA " 2 (  16 MS)"," 2 (  48 MS)"
5360 DATA " 3 (  24 MS)"," 3 (  72 MS)"
5370 DATA " 4 (  38 MS)"," 4 ( 114 MS)"
5380 DATA " 5 (  56 MS)"," 5 ( 168 MS)"
5390 DATA " 6 (  68 MS)"," 6 ( 204 MS)"
5400 DATA " 7 (  80 MS)"," 7 ( 240 MS)"
5410 DATA " 8 ( 100 MS)"," 8 ( 300 MS)"
5420 DATA " 9 ( 250 MS)"," 9 ( 750 MS)"
5430 DATA "10 ( 500 MS)","10 (1.5 S )"
5440 DATA "11 ( 800 MS)","11 (2.4 S )"
5450 DATA "12 (   1 S )","12 (   3 S )"
5460 DATA "13 (   3 S )","13 (   9 S )"
5470 DATA "14 (   5 S )","14 (  15 S )"
5480 DATA "15 (   8 S )","15 (  24 S )"
READY.
```



Basic Min

Een rubriek van Nico Baaijens m

Hoe populair deze rubriek van Commodore Info is geworden, blijkt weer uit het aantal inzendingen. Steeds meer programmerende Commodore-gebruikers isoleren hun routinetjes, truukjes, handigheidjes en maken er een miniatuurtje van, waaraan anderen ook iets hebben. Als zij tenminste even de moeite nemen om het listinkje met kort commentaar en/of uitleg op te sturen naar de redaktie, waarbij liefst linksboven op de envelop CI-miniatuurtje moet staan.

Een typisch voorbeeld van zo'n handigheidje geeft de inzending van H.J. Albers uit Zaandam, die veel met bestanden werkt en een simpele methode vond om getallen van maximaal negen cijfers zonder scheidingstekens rechts aan te lijnen en te voorzien van een decimale punt.

Zijn hulpprogramma is kort en bondig:

```
120 INPUT"GETAL A";A
130 INPUT"GETAL B";B
140 INPUT"GETAL C";C
150 Q=A:GOSUB200
160 Q=B:GOSUB200
170 Q=C:GOSUB200:GOTO300
199 REM HIERONDER DE SUBROUTINE
200 TB=39-LEN(STR$(Q))
210 IF Q=0THEN PRINT TAB(36)"0.00":GOTO250
220 IF Q>0ANDQ<10THEN PRINT TAB(TB-1)Q/100:GOTO250
230IF VAL(RIGHT$(STR$(Q),2))=0THEN PRINT TAB(TB)Q/
   100"00":GOTO250
240 PRINT TAB(TB)Q/100
250 RETURN
300 END
```

Zoals Albers al opmerkt is dit een nuttige routine voor administratieve programma's. Daarmee is hij nogal druk bezig, maar nu komt er een probleem. Al programmerend stuit hij soms op een mysterieuze foutmelding van de C64: "FILE DATA ERROR IN 11050".

Geen enkel Basic- of Commodoreboek vertelt wat die foutmelding precies betekent, laat staan dat er achter te komen is waar de programmeerfout precies zit. Wie geeft het verlossende antwoord op deze brandende vraag?

Basic-uitbreiding

Niet iedereen heeft Simon's Basic, maar waarschijnlijk zal iedereen een nuttige uitbreiding van de standaard C64-Basic op prijsstellen. Het miniatuurtje van F. Buys uit Terneuzen POKEt een stukje machinetaal in het geheugen en vanaf dat moment beschikt de gebruiker over de volgende extra voorzieningen: @W geeft een schoon scherm en is handiger dan (Shift/Clr/Home)

@B,X komt overeen met POKE53280,X

@S,X komt overeen met POKE53281,X

@P,Y,X komt overeen met de opdracht van bijvoorbeeld de TRS-80: PRINT AT X of met MSX- en GWBasic: LOCATE Y,X.

@I,X geeft de inktkleur weer, zoals bij de ZX Spectrum. Hieronder het miniatuurtje:

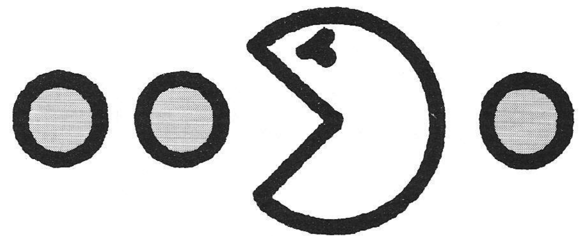
```
100 FORI=49152TO49315
110 READA:POKEI,A:NEXT
120 SYS49152:@W:@B,0:@S,0
130 @I,2:@P,2,12:PRINT "INFO-BASIC 1"
140 END
160 DATA 169,24,141,8,3,169,192
170 DATA 141,9,3,96,32,253,174
180 DATA 32,138,173,32,247,183
190 DATA 166,20,96,234,32
200 DATA 115,0,201,64,240
210 DATA 3,76,231,167,32
220 DATA 115,0,72,32,115
230 DATA 0,104,201,80,208
240 DATA 3,76,87,192,201
250 DATA 73,208,3,76,107
260 DATA 192,201,66,208,3
270 DATA 76,116,192,201,65
280 DATA 208,3,76,125,192
290 DATA 201,83,208,3,76
300 DATA 147,192,201,87,208
310 DATA 3,76,156,192,76
320 DATA 174,167,32,11,192
330 DATA 138,134,251,32,11
340 DATA 192,138,168,166,251
350 DATA 24,32,240,255,76
360 DATA 174,167,32,11,192
370 DATA 142,134,2,76,174
380 DATA 167,32,11,192,142
390 DATA 32,208,76,174,167
400 DATA 32,11,192,165,20
410 DATA 133,251,165,21,133
420 DATA 252,32,11,192,160
430 DATA 0,138,145,251,76
440 DATA 174,167,32,11,192
450 DATA 142,33,208,76,174
460 DATA 167,169,147,32,210
470 DATA 255,76,164,167
```

Nuttig

Bijzonder nuttig is de inzending van Ch. de Vos uit Lokeren, ja alweer uit België. Daar zitten toch maar de echte Commodore-fanaten, die ook weten hoe ze hun miniatuurtjes op de redaktie van Commodore Info en in handen van ondergetekende moeten krijgen.

Dit miniatuurtje is eigenlijk onmisbaar voor iedereen, die programmeert in Basic. De C64 kent geen commando om een per ongeluk geNEWd programma terug te roepen, zoals bij de BBC's, die OLD

iatuurtjes



ɔrte tot zeer korte programma's.

kennen als tegenhanger van NEW om het programma terug te krijgen. Op de TRS-80 onder NEWDOS/80 kan een 'gewist' Basic-programma met RENEW worden terug gehaald. De routine van De Vos geeft, na het invoeren en runnen van het miniatuurtje, elk Basic-programma terug, dat verloren is gegaan door NEW of een RESET met SYS 20000.

```
 50 FORI=0TO44
 60 READA
 70 POKE20000+I,A
 80 NEXTI
 90 END
100 DATA 169,8,141,2,8,76,51,165,172, 174, 0, 140, 45, 0,
    172,175
110 DATA 0,140,46,0,172,174,0,140,47,0, 172, 175, 0, 140, 48,0
120 DATA 172,174,0,140,49,0,172,175,0, 140, 50, 0, 96
```

Teksteffect

Een beter teksteffect dan gegeven in de Miniatuurtjesrubriek van Jaargang 2, nr. 7 geeft Leo J. de Rijk uit Amsterdam. Als dit effect niet de aandacht trekt van de speler van een game of de gebruiker van een nuttig programma, mankeert er iets aan het waarnemingsvermogen van de man of vrouw achter beeldscherm en toetsenbord.

```
100 PRINT "(SHIFT/CLR/HOME)":A$="(37 SPATIES) IK BEN
    EEN TROUWE "
110 B$="LEZER VAN COMMODORE INFO ":C$=A$+B$
130 FORI=1TOLEN(C$):PRINT MID$(C$,I,40):FORJ=1TO-
    100: NEXT:NEXT:RUN
```

Introductiescherm

Ewald van Gemert uit Zeeland (dat schijnt een plaatsje te zijn in Noord Brabant) stuurde wat materiaal op, waarvan de centreerroutine erg bruikbaar kan zijn bij het maken van introductieschermen, die aan programma's voorafgaan.

```
 10 PRINT "CENTREREN"
 20 INPUT"HOEVEEL WOORDEN";W:DIMW$(W)
 30 FORX=1TOW:PRINT "WOORD";X;" ";
 40 INPUT W$(X)
 50 NEXT
 60 INPUT"TEMPO 1 T/M 10";T
 65 INPUT"DOORLOPEND J/N";J$
 70 TE=1000-(T*100)
 80 PRINT "(SHIFT/CLR/HOME)"POKE53280,0:POKE53281,0
 90 PRINT :PRINT
100 FORX=1TOW
110 PRINT TAB((40-LEN(W$(X)))/2)W$(X)
120 PRINT :PRINT
130 NEXT
140 IF J$="N"THEN END
150 GOTO90
```

Crazy Commodore

Het beeldscherm wordt helemaal dol na het runnen van dit graphics-miniatuurtje van Wim Wubs uit Stadskanaal.

```
1 A=53270:PRINT CHR$(142)"(SHIFT/CLR/HOME)"
2 POKEA+11,0:POKEA-5,PEEK(A-5)OR32
3 FORI=1024TO2023-XSTEP2:POKEI,77+K
4 POKEI+1,78-K:POKEA-E,I/8:POKEA+2,21
5 IF INT(I/8)=I/8THEN POKEA+2,Q:Q=Q+1
6 POKEA+10,T:T=-T+1:POKEA+10,9
7 POKEI+54272,1+E:POKEI+54273,1+K:NEXTI
8 E=-E+5:X=500:K=K+1:Q=0:PRINT "(SHIFT/CLR/HOME )":
  GOTO3
```

Handig

Handig vooral is het miniatuurtje van Alfons Uytdehaag (15) uit Roosendaal. Het rekent uit welke POKE-waarden moeten worden gebruikt. De bediening is simpel: teken met het sterretje en druk dan op Return.

```
10 PRINT "(SHIFT/CLR/HOME)"
20 FORT=1024TO1423:POKEX,46:NEXT
30 INPUTA$
40 POKE214,10;SYS58640
50 FORX=1024TO1423
60 IF PEEK(X)=42THEN PRINT X,;
70 NEXT
```

Plotter

Tot besluit een miniatuur van J. Gruteke uit Rotterdam voor de 1520 plotter. Daarover heeft nog nooit iets in Commodore Info gestaan, dus haasten we ons bij deze om dit verzuim goed te maken. Plotteristen zullen dit programmaatje wel kunnen waarderen.

```
10 OPEN4,6:OPEN3,6,3:OPEN6,6,6:OPEN2,6,2
20 INPUT"WAT MOET ER KOMEN TE STAAN ";A$
26 INPUT"GROOTTE LETTERS (0-3)";L:IF L<0ORL>3THEN 26
28 INPUT"BLOK(0)/GEWONE(1) LETTERS";K:IF K<0OR
   K>1THEN 28
30 PRINT "KLEUR 0-ZWART"
31 PRINT " 1-BLAUW"
32 PRINT " 2-GROEN"
33 PRINT " 3-ROOD"
34 INPUTG:IF G<0ORG>3THEN 34
39 PRINT#6,K:PRINT#3,L:PRINT#2,G:PRINT#4,A$: CLOSE
   4:CLOSE3:CLOSE6:CLOSE2
90 PRINT "(SHIFT/CLR/HOME)WILT U DEZE LISTING
   HEBBEN?"
91 GETA$:IF A$=""THEN 91
92 IF A$="N"THEN RUN
93 OPEN4,6:OPEN3,6,3:PRINT#3,"0"
94 CMD4:LIST:PRINT#4:PRINT#3:CLOSE4:CLOSE3:RUN
```

We hebben helaas niet alle bruikbare listings kunnen plaatsen, dus een aantal blijven staan tot de volgende keer. Blijf toch je vondsten insturen, op die manier kunnen we de beste produkten selecteren en daarmee voor iedereen de meeste aantrekkelijke miniatuurtjes plaatsen. Tot de volgende keer.

Nico Baaijens

SOFT WIR WAR

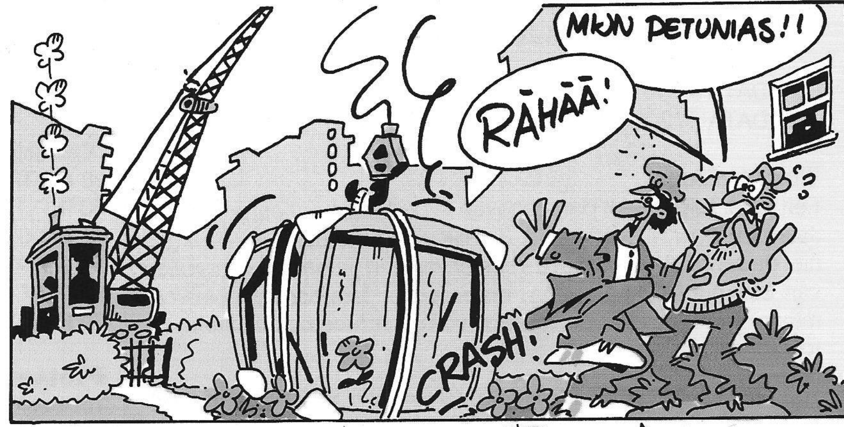
DOOR BERT TIER..









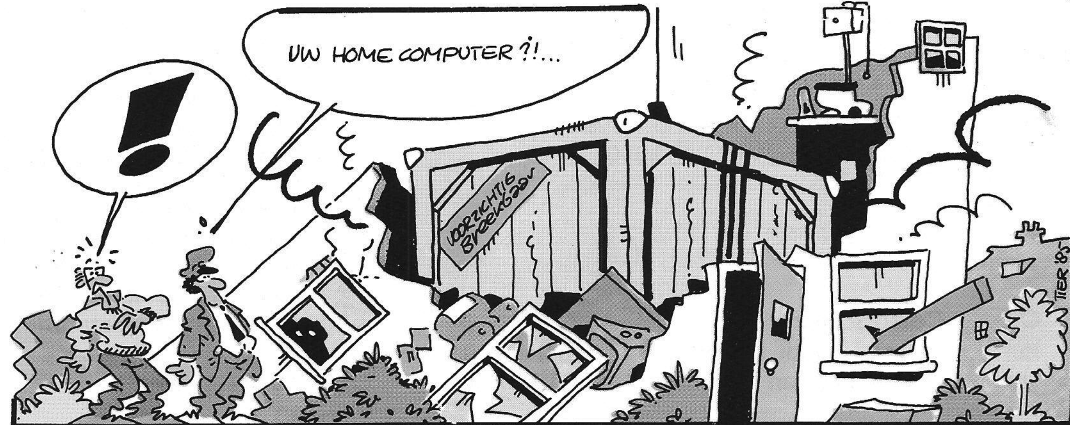














LUC SALA'S DATAKOLOM

Als computerschrijver kun je er natuurlijk niet onderuit, af en toe eens wat te dromen over wat je nu zelf als computerfabrikant wel niet op de markt zou willen. Adam Osborne is daarin het grote voorbeeld, die voegde de daad bij zijn eigen woorden en maakte inderdaad een op dat moment revolutionair computerconcept. Hij maakte de eerste transportabele computer. Bovendien kwam hij op het idee van de bundeling van een breed softwarepakket met de hardware in één koop. Maar ja, zijn ideeën hebben ook hun beste tijd wel gehad, in 1986 vraagt de markt om andere concepten.

Laat ik maar eens een gooi doen naar wat ik denk, dat nu het computergat in de markt is. De specifikatie bedenken voor de machine, waar nu nog een grote afzet voor te vinden zou zijn. Dus niet weer een standaard PC-kloon, maar juist een machine, die binnen die standaard toch anders is. Daarbij ga ik uit van een machine, die helemaal niet bedoeld is voor computerfreaks. Die geletterde computergebruikers, die zwelgen in de nieuwste DOS versies, die denken in bits en bytes en praten over Assembly in plaats van over Assembler, die kopen maar een Amniga of een PC-2 of een ST van Atari.

Wat mij voor ogen staat is een echte boerelullenmachine, die ik dan ook maar de Sala PA 1 wil noemen. Achter deze jolig klinkende naam gaat echter meer schuil dan u misschien denkt, volgens mij is dit het meest onderschatte en onderbediende segment van de computermarkt.

Ik heb het over de gebruikers, die je niet eens naar een cursus durft te sturen, de secretaresses, die niet meer doen dan overtypen, de managers op de vierkante meter, de ambtenaren met werkallergische klachten, het schuim der kantoornatie. U kent ze wel, als ik in de spiegel kijk, zie ik er al eentje.

In de prachtige advertenties in dit en andere computerbladen kom ik nooit een dergelijke doelgroep tegen, maar ik maak me sterk, dat die er toch wel is en voor die groep is dus de Sala PA1.

Dat is qua concept en afwerking werkelijk een perfecte machine, een juweel van degelijkheid zonder een enkele van al die "features" en extra's, waar andere computerfabrikanten zich op laten voorstaan. Laten we er eens een paar onder de loep nemen.

Daarbij gaan we uit van de gebruiker, die op deze machine moet werken, dus niet spelen, er geen cursus computerkunde op hoeft te volgen, van het inwendige niets hoeft te begrijpen en vooral niet afgeleid moet worden door geheimtaal, onbegrijpelijke menu's en procedures.

Om te beginnen het gewicht, dat is hoog, om niet te zeggen zeer hoog. Bij de aflevering van de PA1 valt dat nog wel mee, maar wie eenmaal volgens de installatievoorschriften de apart geleverde verpakkingen zand, grint en cement juist heeft gemengd met wat water en in de daarvoor bestemde opening achterop heeft gegooid, krijgt het ding nauwelijks meer van zijn plaats. Een prima eigenschap, dat beperkt de diefstal en vooral het meenemen van de computer om 's avonds of in het weekend nog even wat te doen.

Voorts kent de PA1 natuurlijk een aantal gebruiksbeperkende voorzieningen. Wat hebben we eraan, als iedereen maar alles met zijn micro kan doen, dat leidt maar tot spelletjes, tot klungelen met veel te ingewikkelde programma's en het illegaal kopiëren van alles en nog wat. Dus de ene diskdrive bevat maar één schijf, waarop geen DOS, slechts een zelfstartend applicatie-programma.

Bij de PA1 zit de DOS in ROM of als het echt niet anders kan in een tweede diskdrive, die dan middels een mooi hangslot is afgesloten.

Bij het afsluiten wordt dan wel automatisch als laatste handeling de diskinhoud van de werkschijf gekopieerd naar de schijf achter het hangslot, dan is het backup-probleem ook opgelost. En de machine uitdoen kan in principe ook niet, want een aan/uit knop zit er niet op. De electronika van de PA1 kan er goed tegen, dat de zaak altijd blijft aanstaan, juist het aan/afzetten geeft problemen. Die DOS werkt dan in basisversie, dus zonder al die mooie extra funkties, geen diskcopy commando, geen mogelijkheid om de **autoexec.bat** *files te veranderen etc.*

De te gebruiken software wordt geleverd met de machine, het zijn toegesneden pakketten, compleet geïnstalleerd zonder de mogelijkheid zelf nog eens allerlei dingen te veranderen. De printeraansturing, als die al nodig is, ligt vast. Die printer is standaard dubbelbreed met 132 kolommen, dat is voor zakelijke toepassingen veel beter en beperkt het maken van de prive-briefjes, een dubbelbrede brief is niet zo eenvoudig te kopiëren. Als er mooie brieven nodig zijn, dan is er vast wel een andere micro met wel die super-faciliteiten en aangekoppelde laserprinter, die dat werkje veel beter kan uitvoeren.

Verder zitten er op de PA1 geen interfaces, zo'n RS 232 is bij gewoon gebruik niet nodig en al die extra's gaan maar stuk of leiden tot verkeerd gebruik. Misschien toch maar een joystick, maar dan wel een, die minstens 30 cm lang is en fysiek de nodige eisen stelt. Op gezette tijden dan toch een spelletje spelen betekent dan tegelijk wat gymnastiek, in de kantooromgeving helemaal niet zo'n gek idee.

U merkt het wel, deze machine is een pure minimum-constellatie. Maar dat blijkt dan ook uit de prijs, voor zo'n 1500 gulden is het ding te maken en op de markt te brengen, met niet meer dan 64 KB, maar wie schrijft er nu in hemelsnaam stukken van meer dan zo'n pagina of 10. ●

Precision Software komt met een nieuw pakket, dat tussen al die andere nieuwe tekstpakketten zeker niet over het hoofd gezien mag worden.

Superscript III.



Na Easy Script voor de CBM 64 heeft Precision Software nu Superscript III op de markt gebracht. Superscript III voldoet aan de vele eisen die computergebruikers tegenwoordig aan tekstverwerkingspakketten stellen en is geheel nederlandstalig. We keken weliswaar naar de engelse versie, maar men is druk bezig met de vertaling, dus die is binnenkort ook wel klaar.

Dus zowel software als handleiding is dan nederlandstalig met alle menu's en commando's in begrijpelijke taal.

Superscript III kenmerkt zich door een grote gebruikersvriendelijkheid en is juist in dat opzicht veel beter dan Easyscript, waar je zonder goede handleiding eigenlijk niet uitkomt.

Menu's

Zodra u het tekst(verwerkings)pakket heeft geladen, verschijnt er direkt een keuze menu. Als opties kunt u nu een werkdiskette, een trainingsdiskette of een woordenboek diskette aanmaken. Heeft u dit al gedaan, dan is het voldoende om de werkdiskette in de drive te plaatsen en op RETURN te drukken.

Het gehele Superscript III programma bevindt zich nu in het geheugen van de computer. U hoeft dus geen diskette's meer te verwisselen voor het verkrijgen van een speciale printer- of tekstverwerkingsfunktie.

De edit-mode is de toestand, waarin Superscript III zich bevindt tijdens het aanmaken en wijzigen van teksten. Zodra u van een van de vele mogelijkheden van Superscript gebruik wilt maken, drukt u op de kommandotoets. Op de bovenste regel verschijnt dan een keuzemenu met verschillende opties. U kunt met de cursortoets het door u gewenste vervolgmenu selecteren. Het kommando dat geinverteerd (verlicht) afgedrukt wordt is het kommando dat geselekteerd staat. Op de regel onder het menu staat nu een korte verklaring van dit kommando. Zodra het juiste kommando geinverteerd staat, drukt u op RETURN om uw keuze definitief te maken.

Snelkiezen

Als alternatief kunt u ook de eerste letter van het gewenste kommando intypen. Dit selecteert direkt het gewenste kommando. Dit zal de ervaren Superscript III gebruiker de mogelijkheid geven net zo snel te werken als hij/zij kan typen.

Natuurlijk zult u wel eens een verkeerde keuze maken. Ook hier heeft Superscript weer een uiterst eenvoudige oplossing. Elk kommando kunt u met CONTROL-Q afbreken!

Vaak wordt gebruikersvriendelijkheid na verloop van tijd als een onprettige en tijdvertragende faktor beschouwt en men dus sneller wil werken dan het worstelen door die behulpzame menu's. Ook hier heeft Superscript III dus een oplossing voor.

Superscript III kan door de onervaren gebruiker d.m.v. de menu's snel en gemakkelijk bedient worden, maar de meer ervaren gebruiker daarintegen zal al snel de eerste letters van de verschillende kommando's gebruiken.

Voor de professionele tekstverwerkings gebruikers, biedt Superscript echter nog een alternatief. Alle belangrijke en vaak gebruikte kommando's kunnen ook via alternatieve toetsen geactiveerd worden. U houd de control toets ingedrukt waarna het bijbehorende kommando wordt uitgevoerd.

Printercommando's

Door de grote verscheidenheid aan printers zijn ook de mogelijkheden om verschillende printer aan te sturen uiterst belangrijk. In Superscript is dit op een zeer eenvoudige en toch doeltreffende wijze opgelost en wel inklusief de seriële aansturing, die ook op de redactie van Commodore-Info Easyscript zo polulair maakte. Zo kunt u met Superscript ook weer de RS232 mogelijkheid van uw computer gebruiken om uw RS232 printer aan te sluiten of om teksten over te sturen (eventueel door middel van een modem!).

Op de Superscript III diskette staan de printerdefinities van de meest gangbare printers zoals Epson en Diablo. Indien uw printer niet als een standaard printerdefinitie op de Superscript diskette staat, kunt u zelf die printerdefinitie wijzigen die het meeste op uw printer lijkt. Zo kunt u dus nagenoeg elke printer gebruiken in samenwerking met Superscript III. Voor gebruikers die over meerdere printers beschikken (een matrix voor het snelle werk en een letterkwaliteit voor het mooie werk), biedt deze methode extra mogelijkheden. Tijdens het werken kunt u namelijk een andere printerdefinitie activeren!

Op uw werkdiskette plaatst Superscript III een defaultbestand. Hierin plaatst u de standaard lay-outwaarden voor uw teksten en geeft u aan welk printertype u gebruikt. Tevens kunt u hier uw eigen tekstverwerkingskommando's en standaardteksten aan een toets toekennen. Zo kunt u bijvoorbeeld met een druk op de toets een standaard document inlezen, of het standaard zinnetje "met vriendelijke groet, M.de Boer" in uw tekst plaatsen. U kunt dus zelf uw tekstverwerkingskommando's samenstellen!.

Rekenen

Nieuw is ook de rekenmogelijkheid van Superscript III. Met Superscript III kunt u nu rekenen als een calculator, (door dus de getallen in te typen) en het resultaat daarna in de tekst te plaatsen.

Staan de getallen echter al in tekst, dan kunt u ook deze getallen gebruiken in uw berekeningen.

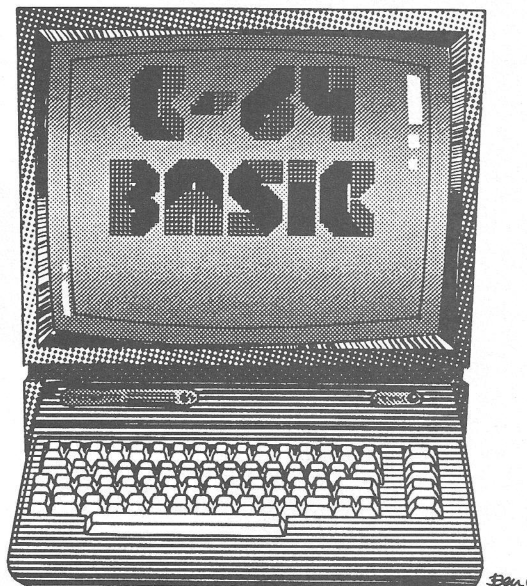
De rekenfunkties bestrijken optellen, aftrekken, delen, vermenigvuldigen en procent berekeningen. Alle rekenfunkties die u bijvoorbeeld voor het maken van een faktuur nodig heeft.

Gaat u echter met getallen en kolommen werken dan heeft Superscript III nog een verassing voor u in petto. De rekenfunktie van Superscript bevat ook opdrachten voor het optellen van de getallen op een regel, het optellen van een kolom en het optellen van meerdere kolommen.

Samenvattend is het duidelijk, dat Precision Software heel duidelijk een gooi wil doen naar een hernieuwde toppositie bij de tekstpakketten. SuperScript lijkt ons het overwegen waard.

Hoewel de C-64 al geruime tijd op de markt is, en er daardoor ook al meer dan 150.000 machines in Nederland zijn verkocht en ook de VIC-20, C-16 en PLUS/4 bezitters samen al een grote groep computeraars vormen, komen er nog elke dag nieuwe gebruikers bij. En zeker nu de eerste C-128 machines de dealers hebben bereikt, zal het aantal computer-bezitters nog groter worden. Voor de beginners gaat Commodore-Info vaak al veel te ver en wordt het te technisch, daarom een serie over Basic voor de nieuwe lezers door Jan Bodzinga, de man die tegenwoordig bij vrijwel iedere computercursus in dit land betrokken wordt.

BACK TO BASIS BASIC



Aan de reakties en vragen van veel van onze lezers merken we, dat er een grote behoefte bestaat aan een betere basiskennis van de standaard gebruikte programmeertaal BASIC.
De meeste Commodore-gebruikers willen toch wat meer doen dan alleen maar het spelen van spelletjes of het werken met voorgeprogrammeerde cartridges. We gaan die groep helpen om met Basic iets meer met hun Commodore-computer te gaan doen. In een aantal afleveringen volgt hier dus een mini-cursus Basic met voorbeelden en uitleg.

> Omdat we beginnen bij de meest algemene serie Basic-opdrachten, maakt het weinig uit over welke Commodore computer je beschikt. Alle hierin besproken Basic-taal kan worden gebruikt bij de hele range van Commodore-computers. We gaan echter uit van de C-64, maar dit is alleen te merken, als er wordt gepraat over het computer-geheugen e.d. want dat is één van de dingen waarin de diverse Commodores nogal verschillen.

Zelf programmeren is niet erg moeilijk, maar het vereist enige moed om te beginnen aan die dikke cursusboeken of in de vaak engelse handleidingen te gaan snuffelen.
Er zijn overigens wel een paar goede redenen te vinden, om eigen programma's te maken en te gebruiken. Zo is het een goede introductie voor algemeen computergebruik, je kunt er programma's mee maken, die er (nog) niet zijn als commercieel pakket en het is ook een leuke bezigheid, die je alleen kunt beoefenen, maar waar je ook met anderen over kunt praten en samen de (helaas onvermijdelijke) problemen oplossen. En met het bezit van een Commodore-computer heb je een prima machine in handen, die erg geschikt is, om zelf te programmeren en met een krachtige en moderne Basic variant. De meeste Basic-opdrachten zijn al zo ingespeeld op het vertalen van je eigen wensen, dat het in feite maar een kleine stap is van het gebruiken van andermans software naar het zelf maken van programma's.

De programmeerladder op

Natuurlijk komt er wel wat voor kijken om zelf te programmeren. Je moet op z'n minst beschikken over wat geduld en uithoudingsvermogen, voordat het eerste zelf geschreven programma kan worden gerund. Maar aan de andere kant is Basic ook weer niet zo moeilijk te leren, dat het een onbegonnen werk is.
Om je behulpzaam te zijn bij het maken van Basic programma's starten we deze serie artikelen die je stap voor stap leert omgaan met de Commodore, de Basic-interpreter, de cassetterecorder, de diskdrive en wat er verder nog bij het maken van programma's komt kijken. We beginnen bij de onderste sport van de programmeerladder, zodat het voor iedereen mogelijk is om met hulp van deze artikelen iets van Commodore en de Basic-taal te gaan begrijpen.

Het begin

Nadat de Commodore computer uit de doos is gehaald en is aangesloten op de monitor of T.V-ontvanger, komt het grote moment, dat de computer wordt aangezet. Als alles volgens de (meestal) nederlandse gebruiksaanwijzing is aangesloten, zal op het scherm een begintekst verschijnen, die aangeeft, dat de computer klaar is, om opdrachten van de gebruiker uit te voeren. Om met Basic te kunnen werken, moet je erom denken, dat alle cartridges zijn verwijderd. Het Basic scherm ziet er bij de C-64 computers uit als volgt:

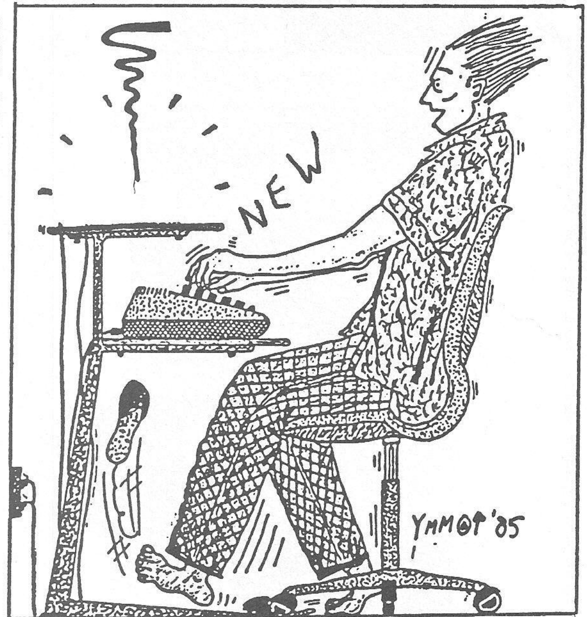
```
*** Commodore 64 BASIC V2 ***
64K RAM SYSTEM 38911 BASIC BYTES
FREE

READY
```

Onder het woord READY zie je een klein, gekleurd, knipperend vierkantje, dat de **cursor** wordt genoemd. Deze cursor laat zien, waar het teken van de volgende toets die je intypt, op het scherm zal verschijnen. Je kunt het beschouwen als een tekstaanwijzer. Als je nu een paar letters op het toetsenbord intypt, zul je zien, dat op het TV-scherm deze letters worden afgedrukt, terwijl de cursor steeds een plek naar

rechts verschuift. Komt de cursor aan het einde van een schermregel, dan verspringt hij naar het begin van de volgende regel op het scherm en gaat vandaar verder.
Het scherm bestaat normaal gesproken, (dus in de tekst-mode) uit 25 regels, die elk 40 tekens kunnen bevatten. In totaal kunnen we dus 1000 karakters tegelijkertijd op het scherm zien.
Deze tekens worden met hulp van programma's, of rechtstreeks vanaf het toetsenbord door de computer naar het scherm geschreven. Naast de gewone cijfers, letters en leestekens, die we kennen van een normale typemachine, vind je op het Commodore-toetsenbord ook nog een heleboel andere, belangrijke toetsen. Deze toetsen hebben stuk voor stuk een bepaalde funktie, die we kunnen gebruiken om er juist die dingen mee te doen, die we wel met een computer, maar niet met een typemachine kunnen doen.

NEW
Dit Basic-commando wordt gebruikt om een programma te verwijderen uit het geheugen van de Commodore. NEW wordt vaak gebruikt voordat met het intypen van een nieuw programma wordt begonnen.



Funktietoetsen

We zullen er een paar onder de loep nemen. Opvallend zijn de 4 extra F-toetsen, die rechts naast het toetsenbord zijn aangebracht. Deze toetsen zijn bedoeld om bepaalde ingewikkelde (samengestelde) opdrachten in één keer aan de Commodore door te geven.
De belangrijkste toets is zonder meer de **RETURN**-toets. Dit is een grote toets, rechts op het middengedeelte van het toetsenbord. Deze RETURN-toets moet iedere keer worden ingedrukt, als je de computer wilt laten weten, dat het intypen van woorden, cijfers of andere gegevens klaar is. De Commodore begint op dat moment de ingetypte informatie te verwerken. Met de **SPATIE**-toets, de brede balk onderaan het toetsenbord, kun je de tekst voorzien van tussenruimtes, zodat niet alle getallen en woorden op het scherm achter elkaar komen te staan. Van de **SHIFT**-toetsen vind je er twee, die beide dezelfde funktie hebben, namelijk het schrijven van hoofdletters in plaats van kleine letters.
Onder de RETURN-toets vind je twee toetsen waarop dubbele pijlen staan aangegeven, met daartussen het woord CRSR. Dit zijn de **CURSOR-TOETSEN**. Met deze toetsen kun je de cursor over het scherm laten bewegen in de richting van de pijl die op de toetsen staat. Gebruik van de toetsen zonder SHIFT zorgt voor een cursorverplaatsing naar onderen (linker toets) en naar rechts (rechter toets). Druk je op één van deze toetsen terwijl je de SHIFT-toets ook indrukt, dan gaat de cursor naar boven of naar links. Deze toetsen zijn erg gemakkelijk, als je b.v. iets verbeteren wilt, wat al op het scherm staat, want tijdens het verplaatsen van de cursor blijft de tekst op het scherm aanwezig en wordt niet verwijderd door de cursor.

De **CLR/HOME** toets zorgt ervoor, dat de cursor naar het begin van het scherm (linksboven) springt, zonder dat de tekst van het scherm verdwijnt. Wil je het hele scherm schoonmaken, dan moet je een SHIFT-toets indrukken en ingedrukt houden, en tegelijk op **CLR/HOME** drukken. Alles wat er op het scherm stond verdwijnt en de cursor staat weer aan het begin van een schoon scherm.
De **INST/DEL** toets heeft ook twee funkties, de eerste (zonder SHIFT) om de cursor 1 positie naar links te verplaatsen en tegelijk het karakter wat daar stond te verwijderen. Gebruik je de INS/DEL-toets met SHIFT ingedrukt, dan zal er op het scherm een plek worden opengemaakt, waar de cursor stond, zodat je in een bestaande regel nog tekst kunt tussenvoegen. De rest van de regel schuift naar rechts en, als hij te lang wordt, zelfs naar de volgende regel.
Om weer een 'schone' goed werkende computer te krijgen, kun je nadat je het nodige op het scherm hebt getypt, natuurlijk de Commodore even uit en aan zetten. Dan krijg je weer de opstarttekst. Een andere (betere) manier is het indrukken van de toetsen RUN/STOP en RESTORE.
Er zitten nog een hele serie andere toetsen op het bord, waar we later nog wel de nodige aandacht aan zullen besteden, voor het ogenblik kun je met de hier beschreven toetsen aan de slag. Probeer maar eens wat teksten op het scherm te typen, en kijk dan wat er gebeurt, als je op RETURN drukt. Je krijgt naar alle waarschijnlijkheid een **ERROR**-melding van de Commodore, maar dat is normaal, want de computer heeft niet begrepen, wat je hebt ingetypt. Om de Commodore te leren kennen, kan het een goede oefening zijn, alle toetsen eens te proberen, en te kijken, wat er op het scherm gebeurt.

Wat is Commodore-Basic ?

De Commodore heeft een eigen taal, Commodore-Basic, die bedoeld is als tussenstap in de communicatie tussen de programmeur en de computer. Het is net als een vakantie in het buitenland. Als je de taal niet spreekt, kom je niet zo ver. Om te kunnen programmeren op de Commodore is het dus nodig om Basic te kunnen lezen en schrijven. Gelukkig is het aantal woorden waaruit Commodore- Basic bestaat beperkt, want meer dan 200 opdrachten en uitdrukkingen zijn er niet. Je zou kunnen zeggen, en daar zit een kern van waarheid in, dat programmeren in wezen niets anders is dan met een heel beperkte woordenschat van steenkolenengels de computer iets duidelijk maken.
De woorden van de Basic-taal leren is dus niet veel moeilijker dan de eerste woordenrijtjes Frans op de MAVO. Maar dat is niet alles. Want bij het maken van een computer-programma komt het niet alleen aan op het gebruik van de goede woorden, maar ook op de volgorde waarin de opdrachten aan de computer worden gegeven.
Voor we verder gaan, nog even iets over de naam BASIC. Basic als naam voor deze computertaal is een aardige vondst. De gevoelswaarde ervan ligt dicht bij de gewone Engelse betekenis van het woord, dat zoiets als basis, fundament betekent, maar daarnaast is het een afkorting van **Beginners All-purpose Symbolic Instruction Code**, wat zoveel wil zeggen als symbolische, allround computertaal voor beginners.
Basic in zijn oervorm is ontwikkeld in de jaren 60 door mensen van het Dartmouth College in de V.S. Oorspronkelijk was het bedoeld om voor beginnende programmeurs een taal te maken, die gemakkelijk te leren is en gebruikt kan worden voor alle dingen die je met een computer kunt doen.

De oorspronkelijke taal bestond uit niet meer dan 50 opdrachten, en is in de loop der jaren enigszins uitgebreid. Met name het bedrijf **MICROSOFT** onder leiding van Bill Gates heeft bijgedragen aan de verspreiding en vervolmaking van Basic. Zij maakten namelijk de eerste versie voor een micro-processor (en daarmee werd de micro plotseling een hanteerbaar ding) en dat is de oerversie geworden van de Microsoft-Basic (**M-BASIC**) in vele varianten. Toen er nogal wat kleine computers op de markt kwamen, die allemaal gebruik maakten van hun eigen Basic-dialekt, duurde het namelijk niet lang of er bestonden talrijke varianten op de oorspronkelijke taal, die onderling weinig meer met elkaar hadden te maken, maar wel allemaal Basic werden genoemd.
Ook voor de Commodore VIC-20 en 64 bestaat zo'n Basic-dialect, door Commodore aangeduid met **V2** ofwel **versie 2.0.** Dit betekent, dat de versie niet zo erg veel afwijkt van de origineel ontwikkelde Microsoft-Basic. Latere computers, zoals b.v. de MSX- serie en de Commodore C-16, PLUS/4 en 128 om er een paar te noemen hebben een **Basic Interpreter** (vertaler) ingebouwd, die veel meer opdrachten kan begrijpen en verwerken. Overigens zijn er ook **Basic-Compilers**, die niet als een Interpreter iedere keer alle commandoregels afgaat en omzet in machinecode, maar dat in één keer doet en een machinetaal programma aflevert. Elders in Commodore-Info wordt over machinetaal genoeg geschreven, dus we keren weer terug naar de Basic Interpreter.

Commodore nummert z'n Basic versies nogal onlogisch met 2.0, 3.0, 3.5, 7.0
Dit geeft in feite allemaal niets, omdat de grondregels voor alle Basic-varianten dezelfde is. En we zullen ons voornamelijk met de basis-principes van Basic bezighouden.

> **Commodore-Basic 2.0 is een versie (ook wel dialekt genoemd) van Basic, die bedoeld is voor alle Commodore VIC 20 en 64 computers. Maar het doet er niet toe of je een Commodore gebruikt met een hoger versienummer, want de latere Basic versies kunnen alle opdrachten van de bestaande versies verwerken. Er zijn alleen méér commando's toegevoegd.**

De dialoog met de Commodore.

Commodore-Basic werkt in principe als een samenspraak tussen gebruiker en computer. Alles wat er op het toetsenbord wordt ingetypt, verschijnt op het scherm, wordt door de in de computer aanwezige tolk (de Basic-interpreter) vertaald en als opdracht uitgevoerd.
Na het aanzetten van de Commodore, waarbij het woord READY op het scherm verschijnt, hebben we twee mogelijkheden, om met de computer te 'praten'. De gemakkelijkste is het direkte gebruik, de **DIREKT**- mode, beter, maar wat ingewikkelder, is het gebruik van programma's (de **PROGRAMMING**-mode). Het verschil tussen deze twee is, dat bij direkt gebruik de opdracht meteen door de computer wordt uitgevoerd, en ook meteen weer is vergeten, terwijl bij het werken met de programma-mode de opdrachten niet direkt worden uitgevoerd, maar eerst in het geheugen van de Commodore worden opgeslagen, tot het woord **RUN** wordt ingetypt. Dan pas worden alle regels van het programma uitgevoerd.
De Commodore moet kunnen uitmaken, of je bezig bent met het ingeven van direkte opdrachten of dat je programma-regels aan het intoetsen bent, die pas later moeten worden uitgevoerd. Het verschil tussen deze twee is simpel, bij direkte opdrachten hoef je alleen de Basic-commando's in te typen (en de RETURN) en voor het invoeren van programma-opdrachten moet elke regel beginnen met een **regelnummer**. Dit regelnummer kan een waarde hebben van 0 tot 63999 en geeft tegelijkertijd de volgorde aan, waarin de programma-regels later moeten worden uitgevoerd.
Omdat de omgang tussen gebruiker en computer over het toetsenbord verloopt, is het logisch, dat voor Basic, naast de feitelijke opdrachten ook leestekens en rekenkundige tekens (operatoren) moeten worden gebruikt. In Commodore-Basic hebben we dus te maken met in totaal 5 verschillende soorten opdrachten:

- **commando's** dit zijn de woorden van de Basic- taal.
- **funkties** dit kun je zien als een verzameling van commando's.
- **opmerkingen** hulpmiddel voor de programmeur.
- **operatoren** wiskundige tekens bv. +,/,- ,*
- **leestekens** zoals bij een typemachine.

De Commodore als rekenmachine.

Om wat met onze Basic-kennis te gaan oefenen, zullen we nu de Commodore in direkt-mode gaan gebruiken als een eenvoudige rekenmachine. Zoals we hebben gezien wordt alles wat we intoetsen en afsluiten met de RETURN toets, door de Commodore bekeken en, als het een juiste (basic)opdracht is, uitgevoerd.

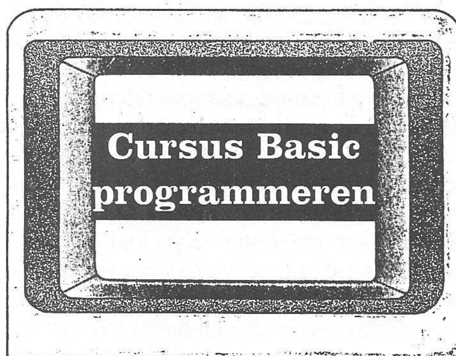
De eerste opdracht die we intypen is :
PRINT 8 (RETURN)
Op het scherm zien we nu deze regel staan, met daaronder een 8. Nog niet erg spectaculair, maar wel degelijk een door de computer uitgevoerde opdracht. Het woord PRINT is een Basic-commando, dat aan de computer vertelt dat er (op het scherm) moet worden geschreven. En wat er moest worden gePRINT is in dit geval het getal 8. De Commodore heeft dit netjes uitgevoerd.
Een stap verder zou kunnen zijn :
PRINT 12 + 7 (RETURN)
Deze regel komt ook keurig op het scherm, met daaronder de uitkomst van de som, dus 19. Zo kunnen we verder gaan, en de meest ingewikkelde rekensommen intoetsen, die daarna door de computer worden berekend en op het scherm gezet.
Probeer maar eens iets als:
PRINT (562 + 38) ★ 15 (RETURN)
en als je dit op een calculator narekent, zul je zien, dat de Commodore zich niet heeft vergist.
De rekensymbolen voor de Commodore wijken iets af van de lagere-school tekens, maar op het moment is het voldoende te weten, dat het vermenigvuldigen op de Commodore wordt aangeduid met een ★', omdat het maalteken teveel lijkt op de

> **RUN**
> *RUN geeft aan, dat we een bestaand programma door de computer willen laten uitvoeren. We kunnen achter het woord RUN een regelnummer typen, waardoor de start van het programma vanaf dit nummer zal plaatsvinden.*
>
> **PRINT**
> *Met PRINT kunnen we getallen, berekeningen en teksten op het scherm zetten. Voor het schrijven van tekst, moeten we de teksten achter PRINT tussen " " plaatsen.*
>
> **PROGRAMMEER-MODE**
> *Dit is de situatie waarin de Commodore zich bevindt als we bezig zijn een programma in te typen. Dit wordt aangegeven, door elke regel die wordt ingetypt, te beginnen met een getal, het regelnummer. Het regelnummer is een positief getal tussen 0 en 65535.*
>
> **DIREKT-MODE**
> *De Commodore is klaar om allerlei Basic opdrachten uit te voeren, direkt nadat ze zijn ingetypt. Deze situatie wordt door de computer altijd gebruikt, als er geen regelnummer als eerste wordt ingetoetst.*

letter 'x'; delingen worden aangegeven met een '/'. Verder worden de ronde haakjes gebruikt om aan te geven welk deel van de berekening het eerst moet worden uitgevoerd. Om dit te kunnen bekijken kun je bovenstaande regel zonder de haakjes laten printen, en dan zie je, dat er ineens een heel andere uitkomst uit de berekening komt. In het eerste geval werd eerst 562 bij 38 opgeteld, en daarna de som met 15 vermenigvuldigd, wat een uitkomst betekent van 9000.
Dezelfde regel zonder haken geeft slechts 1132 als uitkomst op het scherm. Hier zijn eerst 15 en 38 met elkaar vermenigvuldigd en er is daarna pas 562 bij opgeteld.
Deze, op het eerste gezicht zo simpele voorrangsregel is een van de meest voorkomende foutoorzaken in de Basic-programma's. Je zult zeker niet de eerste zijn, die urenlang in een Basic-programma zit te zoeken naar een soortgelijke fout. We komen hier nog wel eens uitgebreid op terug.

Teksten op het scherm.

Om op het scherm een regel tekst te krijgen is niet zo moeilijk. Je maakt het scherm schoon met <SHIFT≥<HOME≥ en begint de tekst op het toetsenbord te typen, waardoor hij ook op het scherm verschijnt. Maar dat is natuurlijk niet wat we bedoelen, want daarvoor heb je de Commodore niet nodig. Laten we het eens proberen met ons PRINT commando.
Logischerwijs moet de Commodore dit begrijpen, want het printen van getallen en sommen is ons ook gelukt.
Probeer maar eens :
PRINT Commodore-INFO (RETURN).
Helaas, er gebeurd wel iets, maar de titel van dit blad komt niet op het scherm. Als het goed is staat er in plaats daarvan een 0 op het scherm. Met andere woorden, dit heeft de Commodore niet begrepen, ofwel wij hebben een fout gemaakt. De oplossing is eenvoudig, want om duidelijk te maken, dat er een tekst letterlijk moet worden overgenomen en op het scherm gezet, moet deze tussen aanhalingstekens ("") worden geplaatst. Als we het nu weer proberen, maar dan op de volgende manier : PRINT "Commodore-INFO" (RETURN), dan zien we, dat het door het toevoegen van aanhalingstekens ineens wel is gelukt.
Op deze manier kunnen we natuurlijk ook getallenreeksen laten verschijnen op het scherm, zonder dat de Commodore de uitkomst al gaat uitrekenen. Bv.:
PRINT "7 + 5 = "; 7+5 (RETURN)
Alles wat tussen " " staat wordt door de Commodore letterlijk afgedrukt op het scherm. Alles wat zonder aanhalingstekens wordt ingevoerd wordt beschouwd als een variabele met een getalswaarde, en als zodanig behandeld.

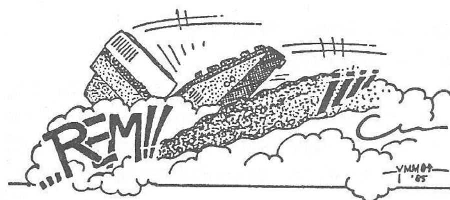


Een Commodore-programma

Bij het werken in direkt-mode zal je opvallen, dat iedere keer dat je de Commodore iets wilt laten doen, je vlak ervoor de opdracht in moet typen. Vooral als het veel dezelfde opdrachten zijn, wordt dit een vervelend en nutteloos werk. Daarom is het beter de commando's in te toetsen in de programma-mode en als het programma dat je hebt geschreven klaar is, kun je in direkt mode RUN intypen en dan pas worden alle opdrachten uit het programma door de Commodore uitgevoerd. Het grote geheim is nu, dat je net zo vaak RUN kunt intypen als je wilt, terwijl je niet steeds weer het programma opnieuw hoeft in te typen. Dat zit al in het geheugen van de Commodore, en door het commando RUN weet de computer, dat hij het aanwezige programma, van af de eerste regel weer moet gaan uitvoeren.

Laten we eens een programma proberen. Zoals je weet kom je (automatisch) in de programmeer-mode van de Commodore door de regel opdrachten te beginnen met een getal.
Dit getal is erg belangrijk, want de Commodore zet de ingetypte getallen in oplopende volgorde in z'n geheugen, onafhankelijk van de volgorde waarin ze worden ingetypt. We noemen het getal, waarmee de programma-regel begint, het regelnummer.
Voor we een programma kunnen intypen, moeten we er zeker van zijn, dat het geheugen van de Commodore geen oud programma of andere gegevens meer bevat, die intern een konflikt zouden kunnen veroorzaken met het nieuwe programma. Om aanwezige data uit het Commodore geheugen weg te halen, typen we de (direktmode) Basic- opdracht NEW (RETURN). De Commodore antwoordt met 'OK' en we weten dat het geheugen klaar is om een programma in op te slaan.
We beginnen het programma door het intypen van:
10 PRINT "HALLO" (RETURN)
Als we dit hebben gedaan, zien we dat de Commodore geen foutmelding geeft op het scherm, dus de ingetypte regel is OK.

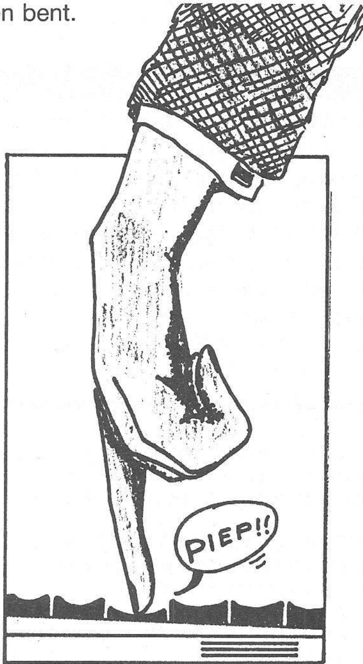
Verdergaand tikken we:
20 PRINT "DIT IS EEN PROGRAMMA" (RETURN)
Vergeet niet na het intoetsen van iedere regel op de RETURN-toets te drukken, het woord (RETURN) moet natuurlijk niet worden ingetypt. Het staat in dit geval alleen achter de in te typen regels om je eraan te herinneren dat je deze toets ter afsluiting moet gebruiken.
We typen nog drie regels, om het programma compleet te maken.
30 PRINT 23 ⋆ 8 (RETURN)
40 PRINT "115 : 5 = " ; 115/5 (RETURN)
50 PRINT "TOT ZIENS" (RETURN)
Nadat je alle regels hebt gekontroleerd op type-fouten kun je het programma laten uitvoeren. Geef de Basic-opdracht :
RUN (RETURN)
en wat gebeurt er.....
Op het scherm verschijnt netjes alles wat we door het intypen van het programma aan de Commodore hebben opgedragen. Eerst wordt in 2 regels de tekst uit de programma- regels 10 en 20 geschreven, dan volgt het getal 184, wat de uitkomst is van de vermenigvuldiging van regel 30 en regel 40 laat naast de tekst ook de uitkomst 23 zien. Ter afsluiting staat er 'tot ziens' uit regel 50 op het scherm. Als we weer RUN ingeven, zien we precies hetzelfde weer verschijnen op het scherm.
Ons eerste programma is gereed. Je kunt nu zelf proberen in de programma-regels andere getallen in te geven, of een andere tekst tussen de aanhalingstekens te plaatsen, waardoor je varianten op dit programma kunt maken. Ook het uitbreiden van dit korte programmaatje zal weinig problemen opleveren, denk ik. Fouten die je maakt kun je verbeteren met hulp van de PIJL-toetsen, waarmee je de cursor over het scherm kunt verplaatsen, maar je kunt ook de hele regel, inclusief het regelnummer opnieuw intypen.
Terwijl we bezig waren, hebben we de eerste Basic-opdrachten onder de knie gekregen.
We kunnen rekenen met de operatoren en getallen, we weten hoe we het geheugen schoon moeten maken en hoe we een programma moeten runnen. Daarnaast kennen we het commando PRINT, zowel voor het typen van tekst als getallen. Ga er zelf mee aan de slag, de volgende keer bekijken we, hoe we leukere dingen kunnen doen, met gebruik van wat meer Basic-opdrachten.
J.B.



UTILITIES 1

Pieptoets

Na het runnen van dit programma krijgt u bij elke aanslag op uw toetsenbord een pieptoon te horen. Dit is heel gemakkelijk bij het intikken van listings. Als u echter een beetje snel tikt zal menig huisgenoot u ervan verdenken dat u met een morsecursus begonnen bent.




```
 1 rem piep-toets / cbm 64
 2 rem door w kok
 3 rem rozenburg
 4 rem
10 fori=0to97:reada:pokei+36864,a:
   next:sys36864
20 data120,169,13,141,20,3,169,144
   ,141,21,3,88,96,169,0,141,0,220
   ,174,1,220,224
30 data255,240,47,169,6,197,251,24
   0,38,169,15,141,24,212,169,128,
   141,6,212,169
40 data131,141,1,212,169,153,141,0
   ,212,169,17,141,4,212,230,251,1
   65,251,201,6
50 data208,5,169,16,141,4,212,76,4
   9,234,169,0,197,251,240,17,230,
   251,169,6,197
60 data251,16,9,169,0,133,251,169,
   16,141,4,212,76,49,234

*** EINDE LISTING ***
```

```
piep-toets

regel 1    228          regel 30   71
regel 2    255          regel 40   221
regel 3    77           regel 50   70
regel 4    143          regel 60   251
regel 10    157
regel 20    72          ready.
```

Sprite-vermeerderen 64

Deze utility geeft u meer dan de 8 standaard sprites op de 64. Voor de mensen die wat gevorderd zijn met hun machine het volgende programma. U zult aan de uitleg in het programma voldoende moeten hebben, daar wij in deze rubriek onmogelijk een cursus kunnen gaan geven. Dus als u meent zowel machinetaal als sprites onder de knie te hebben, dan kunt u het volgende programma eens onder de loep nemen. Een voorbeeld is door Marco Kalter in het programma ingebouwd.

```
  1 rem 16 sprites / cbm-64
  2 rem door marco kalter
  3 rem enschede / 053-314033
  4 rem
 10 poke53281,0:poke53280,0:c=49152
    :print"[CLR-HOME][GEEL][12xspatie]
    een momentje..."
 20 fori=0to352:reada:pokec+96+i,a:
    t=t+a:next:ift=40807thensys4947
    2:goto40
 30 print"er zit een fout in de dat
    aset.":end
 40 print"[CLR-HOME]alle spriteregi
    sters bevinden zich nu[3xspatie]
    vanaf geheugenplaats 49152:"
 50 print"[neer] 0-31 : x en y posi
    ties":print"32,33 : msb's":print
    "34,35 : enable"
 60 print"36,37 : x-expansion":print
    "38,39 : y-expansion"
 70 print"40,41 : priority":print"4
    2,43 : multi color"
 80 print"44,45 : sprite-sprite col
    lision":print"46,47 : sprite-da
    ta collision"
 90 print"48[4xspatie]: sprites 0..
    .7 multi color 0"
100 print"49[4xspatie]: sprites 8..
    15 multi color 0"
110 print"50[4xspatie]: sprites 0..
    .7 multi color 1"
120 print"51[4xspatie]: sprites 8..
    15 multi color 1":print"52-67 :
     sprite colors"
130 print"68-83 : sprite data point
    ers":print"[neer]y-positie spri
    te 0...7 < 130"
140 print"y-positie sprite 8..15 >
    160":print"het programma (her)s
    tart met sys";
150 print" 49472[2xspatie][neer]dru
    k op een toets..":wait197,64,64
    :print"[CLR-HOME]een voorbeeld:"
160 forx=0to7:pokec+1+2*x,50+rnd(0)
    *80:pokec+2*x,(24+40*x)and255
170 pokec+17+2*x,160+rnd(0)*65:poke
    c+16+2*x,(44+40*x)and255:next
180 forx=0to15:pokec+52+x,x:pokec+6
    8+x,255:next
190 pokec+32,192:pokec+33,192:pokec
    +52,6:pokec+42,255:pokec+43,255
    :pokec+48,1
200 pokec+50,10:pokec+49,1:pokec+51
    ,2:pokec+34,255:pokec+35,255
210 data173,34,192,141,21,208,160,7
    ,185,68,192,153,248,7,136,16,24
    7,173,36,192
```

UTILITIES 2

```
220 data141,29,208,173,38,192,141,2
    3,208,173,40,192,141,27,208,173
    ,42,192,141,28
230 data208,173,48,192,141,37,208,1
    73,50,192,141,38,208,160,7,185,
    52,192,153,39
240 data208,136,16,247,173,32,192,1
    41,16,208,160,15,185,0,192,153,
    0,208,136,16
250 data247,173,30,208,141,45,192,1
    73,31,208,141,47,192,169,1,141,
    25,208,169,149
260 data141,18,208,169,208,141,20,3
    ,76,49,234,234,173,35,192,141,2
    1,208,160,7
270 data185,76,192,153,248,7,136,16
    ,247,173,37,192,141,29,208,173,
    39,192,141,23
280 data208,173,41,192,141,27,208,1
    73,43,192,141,28,208,173,49,192
    ,141,37,208
290 data173,51,192,141,38,208,160,7
    ,185,60,192,153,39,208,136,16,2
    47,173,33,192
300 data141,16,208,160,15,185,16,19
    2,153,0,208,136,16,247,173,30,2
    08,141,44,192
310 data173,31,208,141,46,192,169,1
    ,141,25,208,169,255,141,18,208,
    169,96,141,20
320 data3,76,129,234,234,160,95,169
    ,0,153,0,192,136,16,250,169,127
    ,141,13,220
330 data169,96,141,20,3,169,192,141
    ,21,3,173,17,208,41,127,141,17,
    208,169,255
340 data141,18,208,141,25,208,169,1
    ,141,26,208,160,63,185,128,193,
    153,192,63,136
350 data16,247,96,0,0,0,0,0,0,0,4,0
    ,0,4,0,12,42,0,14,43,192,15,255
    ,192,14,42,0
360 data12,32,0,0,168,0,3,187,0,10,
    170,128,14,102,192,10,154,192,1
    4,170,128,10
370 data254,128,14,254,192,10,254,1
    92,10,170,128,2,238,0,0,252,0,0
    ,252,0,0,16,0
380 data0,92

*** EINDE LISTING ***
```

16-sprites

regel 1	184
regel 2	248
regel 3	240
regel 4	143
regel 10	98
regel 20	248
regel 30	143
regel 40	241
regel 50	52
regel 60	223
regel 70	226
regel 80	183
regel 90	216
regel 100	226
regel 110	210
regel 120	215
regel 130	229
regel 140	162
regel 150	135
regel 160	36
regel 170	143
regel 180	34
regel 190	132
regel 200	91
regel 210	26
regel 220	106
regel 230	73
regel 240	253
regel 250	115
regel 260	209
regel 270	84
regel 280	224
regel 290	71
regel 300	47
regel 310	64
regel 320	207
regel 330	217
regel 340	110
regel 350	176
regel 360	209
regel 370	1
regel 380	74

ready.

Extra variabelen

Net zoals bij Sprite Vermeerderen (16 SPRITES) is dit programma bedoeld voor de meer ervaren programmeur. Het zal u misschien dan ook niet verbazen dat het ook door Marco Kalter is geschreven. Ook hier zult u aan de door hem gegeven uitleg voldoende moeten hebben. Mocht u dit programma niet snappen, dan is er geen man overboord. U bewaart het gewoon tot u iets verder bent met uw kennis van programmeren. En als u regelmatig zelf programmaatjes schrijft en deze rubriek doorneemt moet uw kennis vanzelf vooruit gaan. Leren door doen dus.

```
  1 rem extra variabelen / cbm-64
  2 rem door marco kalter
  3 rem enschede / 053-314033
  4 rem
 10 poke53280,0:poke53281,0:print"[CLR-
    HOME][CYAAN]dit programma kan n
    uttig zijn bij het"
 20 print"uitlezen van joysticks en
     het zetten vanx en y posities
    van sprites."
 30 print"[neer][GEEL]sprite positi
    es:":print"[CYAAN]aan de variab
    elen [WIT]x0...x7 [CYAAN]en [WIT]
    y0...y7"
 40 print"[CYAAN]hoeft alleen maar
    een waarde gegeven te worden en
     de bijbehorende ";
 50 print"sprite neemt[2xspatie]die
     positie aan.":print"[neer][GEEL]
    joysticks:"
 60 print"[CYAAN]de variabelen neme
    n vanzelf de juiste[3xspatie]ri
    chtingswaarden aan."
 70 print"[WIT]a [CYAAN]voor poort
    1 , [WIT]b [CYAAN]voor poort 2."
 80 print"[neer]x-richting[2xspatie]
    [WIT]xa,xb[CYAAN]:links-rechts[4xsp
    atie][WIT]-1,+1"
 90 print"[CYAAN]y-richting[2xspatie]
    [WIT]ya,yb[CYAAN]:omhoog-omlaag
    [3xspatie][WIT]-1,+1"
100 print"[CYAAN]vuurknop[4xspatie]
    [WIT]fa,fb[CYAAN]:ingedrukt[7xspati
    e][WIT]-1"
110 print"[neer][l.blauw]het progra
    mma wordt gestart met sys49398[9xsp
    atie][neer]een momentje...."
120 fori=0to431:reada$
130 a=asc(left$(a$,1)):a=a-48+7*(a>
    57)
140 b=asc(right$(a$,1)):b=b-48+7*(b
    >57)
150 v=a*16+b:poke49179+i,v:t=t+v:next
160 ift<>53476thenprint"[op]er zit
    een fout in de dataset.":end
```

UTILITIES 3

```
170 sys49398
180 data a9,00,85,fd,85,fe,c8,b1,2d
    ,29,f0,c9,80,f0,05,a9,02,85,fe,
    60,b1,2d,29,0f
190 data f0,f9,49,0f,aa,c8,b1,2d,30
    ,f1,09,80,85,fe,c8,b1,2d,85,fd,
    46,fe,66,fd,ca
200 data 10,f9,60,a9,01,8d,19,d0,a5
    ,2d,48,a5,2e,48,a0,00,a5,2d,c5,
    2f,a5,2e,e5,30
210 data 90,09,68,85,2e,68,85,2d,4c
    ,31,ea,b1,2d,c9,59,f0,12,c9,58,
    f0,33,a5,2d,18
220 data 69,07,85,2d,90,d8,e6,2e,4c
    ,59,c0,c8,b1,2d,e9,30,90,eb,c9,
    08,b0,e7,0a,48
230 data 20,1b,c0,a5,fe,d0,0a,68,a8
    ,a5,fd,99,01,d0,4c,78,c0,a9,ff,
    85,fd,4c,9a,c0
240 data c8,b1,2d,e9,30,90,c6,c9,08
    ,b0,c2,0a,48,20,1b,c0,68,a8,a5,
    fe,d0,11,ad,10
250 data d0,39,e7,c0,8d,10,d0,a5,fd
    ,99,00,d0,4c,78,c0,c9,01,d0,09,
    ad,10,d0,19,e6
260 data c0,4c,c7,c0,a9,80,85,fd,4c
    ,d6,c0,01,fe,02,fd,04,fb,08,f7,
    10,ef,20,df,40
270 data bf,80,7f,a9,84,8d,14,03,a9
    ,c1,8d,15,03,a5,32,48,a5,31,48,
    a0,00,b1,31,a0
280 data 2a,91,31,a5,31,c5,2d,d0,06
    ,a5,32,c5,2e,f0,0d,c6,31,a5,31,
    c9,ff,d0,e4,c6
290 data 32,4c,06,c1,68,18,69,2a,85
    ,31,68,69,00,85,32,18,a5,2f,69,
    2a,85,2f,90,02
300 data e6,30,a0,00,a9,00,91,2d,c8
    ,c0,2a,d0,f9,a2,00,a0,00,bd,63,
    c1,91,2d,bd,64
310 data c1,c8,91,2d,18,98,69,06,a8
    ,e8,e8,e0,0c,d0,ea,60,58,41,59,
    41,46,41,58,42
320 data 59,42,46,42,00,81,81,00,00
    ,80,29,03,aa,bd,6f,c1,91,2d,c8,
    bd,72,c1,91,2d
330 data 60,ad,01,dc,c5,fb,f0,1b,85
    ,fb,a0,09,20,75,c1,a5,fb,4a,4a,
    a0,02,20,75,c1
340 data a5,fb,4a,4a,4a,4a,a0,10,20
    ,75,c1,ad,00,dc,c5,fc,f0,1b,85,
    fc,a0,1e,20,75
350 data c1,a5,fc,4a,4a,a0,17,20,75
    ,c1,a5,fc,4a,4a,4a,4a,a0,25,20,
    75,c1,4c,53,c0

*** EINDE LISTING ***
```

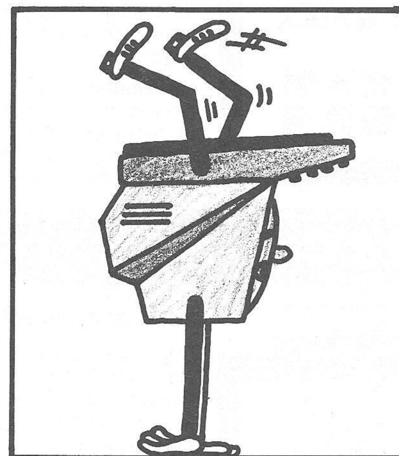
```
extra var.

regel 1     132        regel 80    106
regel 2     248        regel 90    43
regel 3     240        regel 100   47
regel 4     143        regel 110   175
regel 10    155        regel 120   14
regel 20    25         regel 130   166
regel 30    117        regel 140   171
regel 40    113        regel 150   236
regel 50    221        regel 160   19
regel 60    111        regel 170   175
regel 70    25         regel 180   72
```

```
regel 190   162        regel 290   175
regel 200   26         regel 300   11
regel 210   19         regel 310   247
regel 220   67         regel 320   217
regel 230   141        regel 330   65
regel 240   50         regel 340   94
regel 250   47         regel 350   46
regel 260   112
regel 270   231
regel 280   51         ready.
```

Ondersteboven

Met behulp van dit programma wordt al uw tekst onderste boven gezet, een vreemde gewaarwording. Tikt u na het runnen list in dan ziet u dat ook de listing nu ondersteboven staat. Aan de enthousiastelingen die bij het lezen van de eerste zin hun televisietoestel vast gekanteld hebben, nog even de raad om vast een spiegel te pakken daar het ook nog eens in spiegelbeeld staat. Een leuk hulpmiddeltje om bijvoorbeeld een verborgen tekst of de oplossing van iets weg te werken.



```
  1 rem ondersteboven
  2 rem door fer groen
  3 rem uit deurne
  4 rem
 10 print"[CLR-HOME][neer] even ged
    uld a.u.b."
 20 printchr$(142):poke52,48:poke56
    ,48
 30 clr:k=0:dima(511):x=7:y=0
 40 poke56334,peek(56334)and 254
 50 poke1,peek(1)and 251
 60 fori=0to511:a(i)=peek(i+53248):
    next
 70 fori=xtoystep-1:poke12288+k,a(i
    ):k=k+1:next:x=x+8:y=y+8:ifx>=5
    12then 90
 80 goto 70
 90 poke1,peek(1)or 4
100 poke56334,peek(56334)or 1
110 poke53272,(peek(53272)and240)+1
    2
120 print"[CLR-HOME]"
130 rem poke53272,21 zet alles weer
    terug naar normaal

*** EINDE LISTING ***
```

UTILITIES 4

```
onderstebover

regel 1    109          regel 70    103
regel 2    27           regel 80    240
regel 3    68           regel 90    28
regel 4    143          regel 100    193
regel 10    198         regel 110    127
regel 20    236         regel 120    112
regel 30    220         regel 130    154
regel 40    42
regel 50    127
regel 60    118         ready.
```

Diskkeus 64

Het programma diskkeus geeft op het scherm een menu weer met daarin, de op de schijf aanwezige programma's. Nu kunt u door het intikken van een nummer, dat correspondeert met de naam van het programma, dit programma laden. Een heel handige utility, wie altijd zit te modderen met zijn cursortoetsen om achter de gewenste file toch ,8 in te tikken, kent het probleem.

```
  1 rem diskkeus
  2 rem door frank janse
  3 rem uit rhoon
  4 rem
 10 printchr$(147):print"[3xneer][3xspa
    tie]dit programma is gemaakt do
    or :":print
 20 print"[11xspatie]frank janse":print
    :print"[3xspatie]dit is een pro
    gramma dat de in-"
 30 print:print"[3xspatie]houd van
    een diskette leest en":print:print
    "[3xspatie]het in kolom";
 40 print"men op het beeld-":print:
    print"[3xspatie]scherm brengt z
    odat u het dmv":print
 50 print"[3xspatie]het intikken va
    n nummers kunt":print:print"[3xspat
    ie]laden.":forx=0to6000
 60 nextx:dim c$(100):i=0:open1,8,2
    ,"$":b=0:print"[CLR-HOME]"
 70 forv=1to141:get #1,a$:next:j=17
    :gosub230 :n$=b$:j=2:gosub230:i
    $=b$:get#1,a$
 80 j=2:gosub230:o$=b$:forl=1to88:get
    #1,a$:next:print" disknaam:[2xspati
    e]"n$"[2xspatie]"i$:print
 90 forp=1to8:get#1,t$,a$,a$:ift$=""
    thent$=chr$(128)
100 j=15:i=i+1:gosub 230:n$=b$:c$(i
    )=n$:get#1,a$,a$,a$,a$,a$,a$,a$
    ,a$,a$,l$,h$
110 l=asc(l$+chr$(0))+256*asc(h$+chr$
    (0)):if l=0then220
120 ifstthenclose1:goto250
130 b=b+1:ifb=10thenpoke55812,5
140 if b=31 then goto  250
150 ifb=16thenprint"[16xop]":printtab(
    19):print"[ZWART]"b;tab(21):poke
    646,12:printn$:goto210
160 ifb=48thenprint"[18xop]":printtab(
    19):print"[ZWART]"b;tab(21):poke
    646,12:printn$:goto210
170 if b>48 then print tab(19):print
    "[ZWART]"b;tab(21):poke646,12:print
    n$:goto210
180 if b>30then200
190 if b>16 then print tab(19):print
    "[ZWART]"b;tab(21):poke646,12:print
    n$:goto210
200 print"[ZWART]"b;tab(4):poke646,
    12:printn$
210 if p<8 then get#1,a$,a$
220 next p:goto 90
230 b$="":forl=0toj:get#1,a$:ifa$<>
    chr$(96)thenifa$<>chr$(160)then
    b$=b$+a$
240 next:return
250 ifb<30 or b>31then270
260 poke211,0:poke214,21:sys58732:print
    " <<< spatiebalk = volgende pag
    ina >>>"
270 poke211,0:poke214,19:sys 58732
280 print"[2xspatie]tik het program
    manummer in :[7xspatie]"
290 poke211,31:poke214,19:sys 58732
300 gety$:ify$=""then300
310 ify$=chr$(32)then390:x$(1)=y$ :
    printx$(1);
320 x$(1)=y$ :printx$(1);
330 getz$:if z$=""then330
340 x$(2)=z$ :printx$(2)
350 x$(0)=x$(1)+x$(2)
360 x=val(x$(0))
370 ifx<1orx>bthen250
380 load c$(x),8
390 print"[CLR-HOME]":goto150
400 :
410 rem ******** einde ********

*** EINDE LISTING ***
```

```
diskkeus

regel 1    242          regel 210    132
regel 2    166          regel 220    254
regel 3    7            regel 230    230
regel 4    143          regel 240    74
regel 10    245         regel 250    42
regel 20    203         regel 260    5
regel 30    41          regel 270    102
regel 40    166         regel 280    241
regel 50    68          regel 290    154
regel 60    176         regel 300    144
regel 70    113         regel 310    237
regel 80    88          regel 320    57
regel 90    60          regel 330    149
regel 100    182        regel 340    1
regel 110    13         regel 350    86
regel 120    4          regel 360    29
regel 130    207        regel 370    0
regel 140    170        regel 380    7
regel 150    113        regel 390    201
regel 160    152        regel 400    58
regel 170    78         regel 410    148
regel 180    26
regel 190    73
regel 200    47         ready.
```

Print-out

Met o.a. Adresbestand - Telefoon - Rattlesnake - Klutskwijt - Galop 64

Dit is alweer het laatste nummer van dit jaar en we hebben natuurlijk ook nu weer een welgevuld listingdeel.
Voor iedereen, die met de komende feestdagen niet genoeg kan krijgen van het werken met listings, het maken van aanpassingen, verbeteringen, inbrengen van nieuwe snufjes, enz. verwijzen wij naar het zojuist verschenen Grote Commodore-Info Listingboek.
Elders in dit blad staat een kortingsbon om het boek voor de speciale prijs van *f* 19,90 (winkelprijs *f* 24,90) in huis te krijgen.
Schrijft U in uw vrije dagen zelf een programma dat écht het einde of bijna het einde is?
Stuur het naar ons op, wij bekijken het en wie weet, vindt U het terug in één van de komende nummers van Commodore-Info of in een nieuw boek.....

Inhoud van dit listingdeel

Checksum-listtester 64

Nog maar eens het Syntax-checksum programma om uw intikwerk te controleren. Intikken, saven, en dan via RUN onderin in het geheugen plaatsen, dan te testen programma laden (niet runnen) en via SYS 49152 testen.
Dit is de versie, waarbij met Run/stop de listing op het scherm kan worden gestopt en met F-1 of F-7 weer kan worden doorgegaan. Op papier printen gaat met : OPEN 4,4: CMD 4: SYS 49152 en dan na het printen weer PRINT #4: CLOSE 4 om het af te sluiten. Niet alle fouten haalt u hiermee uit uw programma's, want omwisselingen zoals uv ipv. vu komen er niet uit, maar vooral dataregels kunt u er heel goed mee controleren.

```
  1 rem ****************************************
    ******************
  2 rem                basic loader   "SYNTAX.CHECKS
    UM"
  3 rem  runnen en new.  blijft in ram op $c000.
  4 rem           checksum testen met commando  :
    sys 49152'
  5 rem
  6 rem v.840928.64    jan bodzinga
  7 rem ****************************************
    ******************
 10 i=49152  :rem beginadres
 20 reada:ifa<0then50  :rem data ingelezen
 30 pokei,a:i=i+1:goto20
 50 poke49184,148:poke49185,192
 55 i=49300
 60 read a: ifa<0then90
 70 pokei,a:i=i+1:goto60
 90 print"data is weggezet"
 95 print"checksum testen met sys49152"
100 data 165,  43, 166,  44, 133, 163, 134, 164,
    169, 147
110 data  32, 210, 255, 160,   0, 240,   3,  32,
     73, 192
120 data  32,  73, 192, 208,   1,  96,  32, 225,
    255, 208
130 data   3,  76, 116, 164,  32,  81, 192,  32,
     73, 192
140 data 240,  12, 201,  32, 240, 247,  24, 101,
    167, 133
150 data 167,  76,  37, 192, 166, 167, 169,   0,
    132, 168
160 data  32, 205, 189, 169,  13,  32, 210, 255,
    164, 168
170 data  76,  17, 192, 200, 208,   2, 230, 164,
    177, 163
180 data  96, 162,   0, 189, 123, 192, 240,   6,
     32, 210
190 data 255, 232, 208, 245,  32,  73, 192, 170,
     32,  73
200 data192, 132, 168,  32, 205, 189, 162,   3, 1
    69,  32
210 data  32, 210, 255, 202, 208, 250, 169,   0,
    133, 167
220 data 164, 168,  96,  82,  69,  71,  69,  76,
     32,   0
230 data  -1
240 data 165,197,201,3,240,7,201,4,240,6
250 data 76,148,192,76,34,192,169,147,32,210
260 data 255,76,161,192
270 data -1

              *** EINDE LISTING ***
```

Checksum: Checksum.nw

regel	1	249	regel	110	158
regel	2	84	regel	120	232
regel	3	6	regel	130	183
regel	4	117	regel	140	96
regel	5	143	regel	150	96
regel	6	17	regel	160	127
regel	7	61	regel	170	71
regel	10	157	regel	180	223
regel	20	65	regel	190	73
regel	30	203	regel	200	79
regel	50	14	regel	210	109
regel	55	251	regel	220	106
regel	60	193	regel	230	225
regel	70	207	regel	240	114
regel	90	245	regel	250	89
regel	95	237	regel	260	68
regel	100	183	regel	270	225

Galop 64

Een paard in beweging, dat vraagt behoedzaam manoevreren en deze animatie vinden we dan ook wel geslaagd. Even geduldig intikken en dat galoppeert er een paard over het beeld.

```
  1 rem galop / commodore-64
  2 rem door m steenbergem
  3 rem vlaardingen / 010-743166
  4 rem
 40 poke 53280,2:poke 53281,2
 50 print"[CLR-HOME]"
 60 v=53248:poke v+21,7:poke 2040
    ,13:poke 2041,14:poke 2042,15
 70 for i=832 to 1024:read n:poke
    i,n:print i:next i
 80 poke v+23,7:poke v+29,7
 90 poke v+39,0:poke v+40,0:poke
    v+41,0
100 poke v+0,100:poke v+1,100
110 print"[CLR-HOME]"
120 for g=0 to 3
130 for x=0 to 255:poke v+0,x
140 if int(x/5)=x/5 then read d:poke
    2040,d
150 next x
160 next g
170 data 0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,30,0,0,62,0,124,255
    ,1,255,243,3,255
180 data 240,6,255,240,12,255,224
    ,0,239,224,1,231,192,1,193,19
    2,2,129,64,2,130
190 data 64,4,140,64,0,0,64,0,0,6
    4,0,0,32,0
200 data 0,0,0,0,0,0,0,0,96,0,0,2
    40,0,1,240,0,3,248,0,255,200,
    3,255,192,2,255
210 data 128,4,255,128,12,255,128
    ,8,127,0,0,55,128,0,51,64,0,5
    1,32,0,49,64,0
220 data 31,192,0,12,0,0,8,0,0,4,
    0,0,0,0
230 data 0,0,0,0,0,0,0,0,0,96,0,0
    ,240,0,1,240,0,1,248,0,199,20
    0,1,255,128,7
240 data 255,128,15,255,128,25,25
    5,128,49,255,128,1,223,192,0,
    194,64,0,193,64,0
250 data 229,64,0,147,64,0,136,64
    ,0,68,0,0,64,0,0,32,0,0,0,0
260 data 13,14,15,14,13,14,15,14,
    13,14,15,14,13,14,15,14,13,14
    ,15,14
270 data 13,14,15,14,13,14,15,14,
    13,14,15,14,13,14,15,14,13,14
    ,15,14
280 data 13,14,15,14,13,14,15,14,
    13,14,15,14,13,14,15,14,13,14
    ,15,14
290 data 13,14,15,14,13,14,15,14,
    13,14,15,14,13,14,15,14,13,14
    ,15,14
300 data 13,14,15,14,13,14,15,14,
    13,14,15,14,13,14,15,14,13,14
    ,15,14
310 data 13,14,15,14,13,14,15,14,
    13,14,15,14,13,14,15,14,13,14
    ,15,14
320 data 13,14,15,14,13,14,15,14,
    13,14,15,14,13,14,15,14,13,14
    ,15,14
330 data 13,14,15,14,13,14,15,14,
    13,14,15,14,13,14,15,14,13,14
    ,15,14
340 data 13,14,15,14,13,14,15,14,
    13,14,15,14,13,14,15,14,13,14
    ,15,14
350 data 13,14,15,14,13,14,15,14,
    13,14,15,14,13,14,15,14,13,14
    ,15,14
360 data 13,14,15,14,13,14,15,14,
    13,14,15,14,13,14,15,14,13,14
    ,15,14
370 data 13,14,15,14,13,14,15,14,
    13,14,15,14,13,14,15,14,13,14
    ,15,14
380 data 13,14,15,14,13,14,15,14,
    13,14,15,14,13,14,15,14,13,14
    ,15,14
390 data 13,14,15,14,13,14,15,14,
    13,14,15,14,13,14,15,14,13,14
    ,15,14
400 data 13,14,15,14,13,14,15,14,
    13,14,15,14,13,14,15,14,13,14
    ,15,14

*** EINDE LISTING ***
```

```
galop

regel 1    109        regel 220   62
regel 2    65         regel 230   222
regel 3    226        regel 240   12
regel 4    143        regel 250   37
regel 40    41        regel 260   171
regel 50    112       regel 270   171
regel 60    32        regel 280   171
regel 70    72        regel 290   171
regel 80    254       regel 300   171
regel 90    130       regel 310   171
regel 100    67       regel 320   171
regel 110    112      regel 330   171
regel 120    129      regel 340   171
regel 130    128      regel 350   171
regel 140    48       regel 360   171
regel 150    218      regel 370   171
regel 160    201      regel 380   171
regel 170    70       regel 390   171
regel 180    185      regel 400   171
regel 190    229
regel 200    67
regel 210    84       ready.
```

Handwerk 64

Om letters op het scherm te krijgen in afwijkende vormen is er dit programma. Let er op, eerst even POKE 44,64 en POKE 16384,0 doen voor het geladen wordt. De letters komen vanzelf op het scherm, maar wie andere tekst wil, moet gaan studeren op de dataregels.

```
 1 rem handwerk   /  cbm-64
 2 rem door erik scheffers
 3 rem 's-hertogenbosh
 4 rem
10 ifpeek(44)<>64thenprint"typ e
   erst poke 44,64:poke 16384,0
   met[3xspatie]return en ";
20 ifpeek(44)<>64thenprint"laad
   dan dit programma !":end
30 base=2*4096:print"[CLR-HOME]e
   ven geduld (30 seconden)"
```

Vervolg op pag. 55

Een wondermachine, bericht men ons uit Amerika en ook de paar softwarehuizen hier, die er al mee werken, zijn zeer enthousiast over de brede mogelijkheden van de Commodore Mac-eter. In München speelden we zelf een tijdje met de machine en waren verrast door de brede mogelijkheden, maar hebben toch wat aanmerkingen.

DE AMIGA NADER BEKEKEN



In een eerder nummer (7) van Commodore Info besteedden wij al uitgebreid aandacht aan de specificaties van de nieuwe AMIGA en gaven de impressies van Maureen Fleming, vers uit de VS. Nu wordt het tijd om eens een blik in het inwendige van deze geavanceerde PC te werpen en een aantal mogelijkheden eens nader te bekijken. Wellicht valt de vraag of de AMIGA de "droommachine" is, dan te beantwoorden.

Om te beginnen viel ons bij het gebruik op, dat al de mooie graphics en geluiden wel erg veel tijd vragen om ingeladen te worden. Het duurt minuten om zo'n prachtige demo van schijf in te laden, zelfs bij een machine, die omringd wordt door 2 stuks 3,5 inch drives en een extra 5 inch eenheid. De processor is snel genoeg, de extra chips maken echt een nieuwe generatie software mogelijk, maar de I/O is duidelijk te traag. De vrijheid om alles met uiterst gedetailleerde bit-mapped graphics uit te voeren, wordt door sommige ontwerpers wel ten volle benut, maar de randapparatuur kan het niet bijsloffen. Eigenlijk is de Amiga qua chips en snelheid een 1985 machine, die wordt geremd door randapparatuur en een SASI interface uit het begin van de tachtiger jaren. Nieuwere en vooral snellere diskdrives, hard-disks met 40 tot 100 MegaByte zijn de randapparaten, die hierbij passen. Ongetwijfeld komen die nog, maar dan is op dit moment een Amiga met 256 KB en een 3,5 inch drive nog wel een wat kreupel geval in dit opzicht. Verder, als het programma en de software er eenmaal inzit, werkt het wel fraai, snel en soms uniek mooi. Dat men daarbij zo beknibbelt op het vrije geheugen is onbegrijpelijk, met 256 KB doe je niet zo veel, zeker niet als de HI-RES graphics mode per beeldje 160 KB videogeheugen verslindt.

MS-DOS brug

Commodore heeft wel ingezien, dat de zakelijke wereld nooit een PC zal omarmen, die geen weg biedt naar de nu overal gebruikelijke MS-DOS standaard. Daarom heeft men bij de introductie gezegd, dat er daarvoor een speciale kaart en interface zou komen. Daar wachten we met spanning op, want dat zou de Amiga in een klap tot echte concurrent van de PC- compatibles maken.

Systeem-architectuur

Al het rekenwerk in de AMIGA wordt door drie verschillende processorchips naast de 68000 afgehandeld. Die zgn. Amigachipset is de geheime kracht van het systeem.

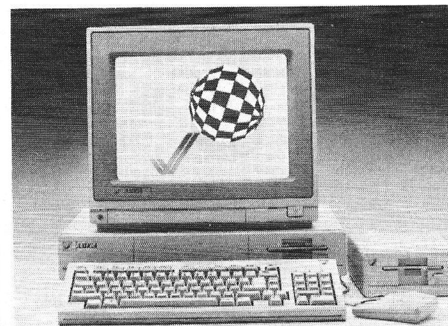
Daphne verzorgt de beeldscherm opbouw in een of twee onafhankelijke beelden, in 1 tot 4096 kleuren en controleert daarbij ook nog acht drie-kleurige sprites.

De **Agnus** chip bestaat in feite uit twee delen: De "Copper" en de "Blitter".

Copper staat voor de coprocessor die de graphics- en geluidschips stuurt. Copper runt zijn eigen programmaatje, afhankelijk van de positie van de elektronenstraal, die het videobeeld opbouwt. De lokatie van de elektronenbundel bepaalt welk dataregister Copper gaat verwerken. Voor het snel uitlezen van de registers gebruikt deze coprocessor D(irekt) M(emory) A(cces). Er zijn maarliefst 25 DMA-kanalen waarlangs "Copper" direkt andere chips kan bereiken.

Blitter staat voor Bit Mapped Block Transfer en trekt lijnen, vult de verschillende kleurgebieden in en kan rechthoekige pixel (=beeldpunten)-blokken verplaatsen. De "Blitter" is verantwoordelijk voor de indrukwekkende grafische mogelijkheden en truckages van de AMIGA.

De laatste van de superchip drieling is **Portia**, een I/O chip voor het poortwerk. Portia houdt zich bezig met de in/output, de vier geluidskanalen, interruptcontrole, diskettebesturing, de communicatie via interfaces, en de gegevens van muis of joystick. Kortom een echte poortwachter.

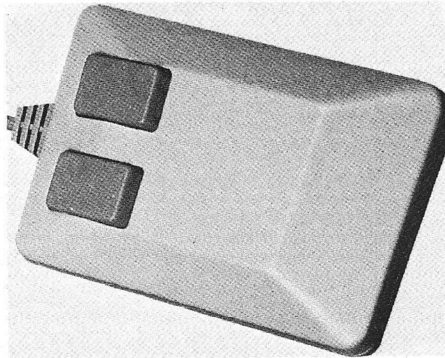


Taakverdeling

In de AMIGA zijn de taken dus duidelijk verdeeld. Deze taakverdeling ontlast de **68000 hoofdprocessor** van de Amiga en levert daarbij een flinke tijdswinst op. Commodore claimt dat de AMIGA drie maal zo snel is als de Mac van Apple. Het is daarbij wel jammer, dat men heeft vastgehouden aan de 68000 en geen 68010 of 68020 heeft gebruikt, waar men de datastromen zonder multiplexing 16 of 32 bit breed mee kan verwerken en niet moet wachten op de afhandeling in de **Wait** state. Het is eigenlijk een vreemde situatie, men neemt een supersnelle chip, maar gebruikt die capaciteit dan wel voor veel bit-map werk en luxe gebruiksvriendelijkheid, maar dan zijn er weer extra versnellers nodig.
Belangrijk is echter wel, dat de AMIGA door taakverdeling aan meerdere programma's of programmaonderdelen tegelijk kan werken. Wij spreken hierbij van **Multitasking**, waarbij dus bv. een afdrukjob, tekstverwerking en sorteren van een file tegelijkertijd kan plaatsvinden. Het verloop van de verschillende programma's is eventueel via elkaar overlappende **windows** te volgen.

Multitasking De AMIGA leent zich uitstekend voor het draaien van meerdere toepassingen tegelijk. Zojuist hebben wij al gezien dat de taken verdeeld zijn, zodat de 68000-processor zoveel mogelijk beschikbaar blijft. Dat is echter nog niet voldoende voor een snelle multitasking machine. De AMIGA heeft daarom nog meer in zijn mars. Zo beschikt de 68000-processor over een **EXEC-systeem.** EXEC staat voor een uitgebreide collectie ROM-routines waarmee snel een aantal vitale functies verricht kunnen worden. Bijvoorbeeld het organiseren van device in/output, het afhandelen van interrupts en het maken van **list nodes.** Een "list node" is een manier van databeheer met pointers naar voorafgaande-, opvolgende- en geassocieerde datablokken, zodat die sneller ter beschikking zijn.

Behalve de EXEC-routines zijn er ook nog de zgn. **LIBRARIES.** Deze bibliotheken zijn er om nuttige routines, met behoud van maximale flexibiliteit, zeer snel te kunnen organiseren. Een Library kan op elk geheugen-addres worden ingezet en met een zelf gemaakte code worden opgeroepen. Als laatste de **interrupts, de signalen van buiten, die de processor tot andere taken manen. Bij de Amiga zijn die via de randchips zo veel mogelijk buiten de deur gehouden.** De 68000 trekt zich helemaal niets van de perifere chips aan totdat een van hen een interruptsignaal uitzendt. Bovendien handelen deze randchips veel in



De muis is een belangrijk attribuut voor de grafische commando's van de Amiga.

terrupts van buiten (bijvoorbeeld de diskdrive) zelfstandig af en raadplegen de hoofdprocessor alleen voor werkelijk belangrijke zaken.

Samen op de bus

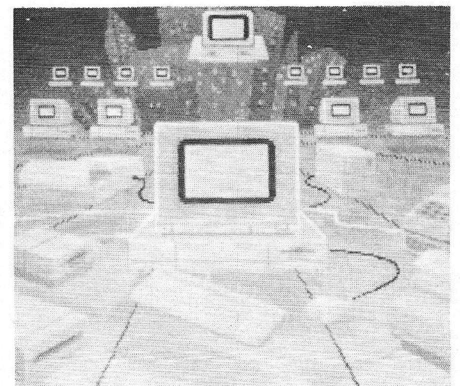
De AMIGA kan vele functies tegelijk verrichten zonder dat de 68000-processor noemenswaardig vertraagd wordt. Bijvoorbeeld met de diskdrive communiceren, acht sprites over het scherm jagen, terwijl ze voortdurend van kleur veranderen, en het geheel nog even muzikaal omlijsten via de vier audiokanalen.
Dit is eigenlijk alleen maar mogelijk doordat al de verschillende subsystemen samen de systeembus delen. In overvolle bussen is het echter slecht vertoeven en menigeen mijdt daarom liever de spitsuren. Zo ook bij de AMIGA. De 68000 chip heeft een **kloksnelheid van 7,15909 MHz,** terwijl het geheugen ongeveer twee maal zo snel is. De meeste instructies in de 68000 alterneren tussen bus gebruik en interne verwerking. Daardoor kan het geheugen op topsnelheid lopen en toch elke andere buscyclusperiode vrij laten.
De gebruikscycli zijn afhankelijk van de tijd die het elektronenkanon van de monitor of tv nodig heeft een een pixellijn te trekken en te controleren. Dat kost c.a. 63 microseconde. Deze wordt opgedeeld in 226 memory access cycli van 280 nanoseconde elk.
Copper en Blitter gebruiken de even cycli. De overige cycli zijn voor Sprite-, Memory Refresh-, Audio-, Disk-DMA enz.

Graphics

Zoals al vele testen uit volle borst verkondigd hebben: de grote kracht van de AMIGA schuilt in de formidabele graphics. De specificaties van een 640x400 pixel-resolutie in 16 kleuren en met controle van 8 multicolor sprites is al zeer goed, hoewel sommige speciale PC kaarten in de buurt hiervan komen. De Blitter-activiteiten slaan echter alles wat men tot nu toe op de PC-markt vertoond heeft. De AMIGA werkt met zgn. bit maps. Dat zijn rechthoekige pixel-blokken in het geheugen. Met deze bitmaps bouwt de computer het beeldveld op. Ook een sprite is een stukje bitmapbeeld. Het bewegen gebeurt eenvoudig door de sprite-positie aan te geven.
De Blitter verplaatst zoals de naam al aangeeft bit map-blokken. Aangevuld met een groot aantal ROM-animatieroutines presteert de AMIGA op animatiegebied meer dan andere PC's en vele main frames.

Gels

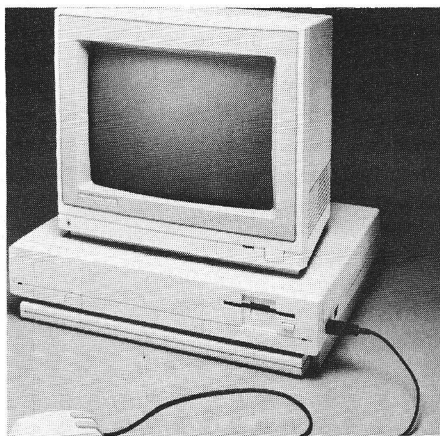
Het basiselement in de animatiesubroutine is de **GEL** of grafisch element. Er zijn vier GEL-typen:
De V- of **Virtual sprite.** Dat is een data structuur in het geheugen, die nauw met de eerder besproken hardware sprites verbonden is en door de animatieroutines "gemapt" wordt. BOB's of Blitter Objects is een beeld dat als een sprite opereert, maar de animatieroutine gebruikt de Blitter om het beeld op het beeldveld te plakken en (naar keuze) het beeld dat onder de BOB lag terug te plaatsen. In praktijk maken de BOB's het mogelijk om met grafische voorstellingen te manipuleren. Een tekenfilmpje maken bijvoorbeeld kan nu thuis met de AMIGA, indien u tenminste genoeg interne en externe opslagcapaciteit voorhanden heeft, want bit-mapped graphics kosten veel geheugen. Een wandelend mannetje heeft bijvoorbeeld BOB's voor armen, benen, hoofd en torso. Deze kunnen dan afzonderlijk of als een geheel bewegen.
Een combinatie van BOB's noemt men een **AnimObj.** Een deel van een AnimObject heet een AnimComp(onent). Respectievelijk GEL-type drie en vier. Behalve verschillende animatievormen zoals motion-controll- of sequenced drawing animation (de computer berekent de verschillende bewegingsfasen aan de hand van de ingevoerde waarden) zijn er nog een groot aantal trucages mogelijk. Voorwerpen en teksten kunnen in elkaar overgaan, kleuren veranderen, perspectieven verschuiven, noem maar op.



De grafische mogelijkheden van de Amiga blijven verbluffend.

Gebruiksvriendelijkheid

De ontwerpers van de AMIGA hebben veel aandacht aan de gebruiksvriendelijkheid van hun schepping besteed. **Intuition** heet het gebruikersinterface dat DOS, muis en Ikonen (pictogrammen of gebruikersplaatjes) bestuurt. Ook de aparte programmawindows vallen onder **Intuition.** Voor de video-output gebruikt Intuition screens. Deze schermen zijn rechthoekig, even breed als het beeldscherm, bezitten een vaste kleurkombinatie en hoogte, en een identificatiebalk. De screens hebben **pull-down menu's**, d,w.z. dat u met de muis de menu's voor de in het desbetreffende window lopende programma kunt opvragen door het trekmenu zichtbaar te maken.
De AMIGA biedt diverse soorten windows. waaronder zelfs met scrollende inhoud en extra geheugencapaciteit voor het grotere (grafische) werk.
Ook biedt Intuition een **"Workbench"** (=werkbank). Het werkbankscherm opent automatisch als een disk die de benodigde software bevat, geladen wordt. Ook werkbank is een op windows gebaseerd commando-interface. Het is zowel een screen



waarop de gebruiker diskettes , tools, drawers, clipboards (prikbord) en de vuinisbak kan openen terwijl de andere programma's lopen. Het openen van een **"Tool"** (=programma) creëert meteen een window op het huidige scherm. **Drawers** (laden) zijn ikonen die net als een window een bepaalde inhoud bevatten. U kunt echter zelf naar believen items in de la leggen of er uithalen. Een clipboard maakt het de gebruiker mogelijk om snel informatie tussen de verschillende programma's of projecten over te dragen.

De AMIGA is in zijn klasse een zeer veelzijdig apparaat met een interrassante systeemarchitectuur en vele mogelijkheden. Een ware droommachine voor de hobbyist(e) met noten op zijn of haar zang of het kleinbedrijf met specifieke grafische wensen. Helaas zal de prijs van meer dan f 7000,- voor een volledig opgetuigde versie de gemiddelde hobbybeurs te boven gaan. Met de beloofde MS-DOS brug erbij, die we voor zakelijk gebruik esentieel achten, komt men nog duurder uit. Toch zijn wij wel gelukkig met de AMIGA, die een nieuwe generatie vormt qua gebruiksvriendelijkheid, snelheid, graphics en toepassingmogelijkheden verbeteren. Of dergelijke kwaliteiten voldoende zijn om de computermarkt weer wat op te vijzelen, we wachten maar even af. ●

Hieronder staan een aantal poke-kommando's die samen met een aantal spelen gebruikt kunnen worden. Het resultaat van zo'n poke moet je zelf maar bekijken. Ze liggen in de trant van: meer levens, geen vijanden meer, snelheid lager of hoger, je kunt niet meer dood enz... enz...

SUPERPOKE'S VOOR SPELFANATEN

Pokes voor de Commodore 64 voor diverse spelen.

De poke-kommando's zijn door meerdere mensen ingestuurd, bij elkaar gezocht en voor een deel getest, maar niet allemaal met garantie, soms zijn er Pokes, die niet op alle ROM versies werken etc. Het terrein van de geheime spel-pokes, die intern dingen veranderen, is nog veel groter, dus we komen in de toekomst nog wel meer met zo'n lijstje.

Manic Miner	53264,126
Dragon Hawk	3477,255
Frantic Freddie	31887,255
Ghouls	8367,255 of 31887,50 of 8367,50 of 34535,24
Jumpman jr.	9450,173
Moon Buggy	30624,173 of 9450,13 of 24151,173
Galaxians	7065,230
Burning Rubber	18432,173 of 17288,165
Pooyan	20634,173
Q-Bert	4446,173
R-Nest	2759,100
Sammy Lightfoot	3678,183
Hart Hat Mack	16877,173 of 3678,255 of 8472,100
Battle trough T	22045,255
H.E.R.O.	14652,25
Super Scramble	4691,X
Punchy	15458,50
Manic Miner	3574,44
Cuthbert Jungle	2659,5 of 5989,58
Protector II	16425,6
Encounter	30430,0
Skramble	11291,175
Spelunker	10407,44
Son of Blagger	6626,232
Defender	3005,5 of 3560,8
Congo Bongo	3655,5 of 39132,48
Buck Rogers	2490,9
UGH!	22178,255 of 8825,36
Neptunes Daughter	7870,60
Alligata Blagger	53264,126 of 9521,234 of 3574,44 en 9522,234
Mr.Robot	11518,255 en 9523,234
Surviver	19563,255
Evolution	6947,255 of 19523,200
Bagitman	22236,255
Cavelon	23789,255
High Noon	18033,255 of 15458,255
Flak	4798,36
Loderunner	
Galaga	17788,165 en 17383,173 en 17388,173
Hunchback	9521,234 en 9522,234 en 9523,234
	of 22521,44 of 7870,60 of 9521,44 of 5704,50
Arabian Nights	2631,173 en 2632,141 en 2633,169 en 2634,89
Loderunner ROM	7892,255
Miner 2049'er	2652,165 of 2471,255
Jungle Hunt	2242,234 en 2243,234
Kid Grid	10020,234
Robin Rescue	6144,234
Choplifter	8011,173
Ring of Power	43,207 en 44,24
Gateway Apshai	2264,99
Lancer Lords	16424,60
Time Runner	8543,9

Veel speelplezier met deze pokes, die je dus na het laden en voor het runnen moet ingeven, waarbij de dus niet de Autorun methode kunt volgen. ●

Vervolg van pag. 46

```
 40 fori=basetobase+7999:pokei,0:
    next.
 50 poke53265,peek(53265)or32:poke
    53270,peek(53270)or16
 60 poke53272,peek(53272)or8
 70 poke53280,0:poke53281,0:diml$
    (255)
 80 readl,l$:ifl$="**"then110
 90 ifl=othenl$(l)=l$(l)+l$:goto8
    0
100 l$(l)=l$:o=l:goto80
110 c=1:sp=1:tx$="[2xspatie]ZELFS
    CHRIJVENDE▲[6xspatie]LETTERS▲
    ▲Door Erik Scheffers▲"
120 gosub190
130 tx$="▲U kunt zelf de ":gosub1
    90:c=5:tx$="kleur":gosub190
140 c=1:tx$="veranderen ":gosub19
    0
150 tx$="en ook▲de ":gosub190:sp=
    100:tx$="snelheid.▲":gosub190
    :sp=1
160 tx$="":fora=32to95:tx$=tx$+chr$
    (a):next
170 fora=192to223:tx$=tx$+chr$(a)
    :next:gosub190
180 goto180
190 fora=1tolen(tx$):z=asc(mid$(t
    x$,a,1)):ify>200thenreturn
200 ifz=186orz=13thenx=0:y=y+8:if
    y>200thenreturn
210 ifz=186orz=13thennext:return
220 tc$=l$(z):forb=1tolen(tc$)step
    2:xp=val(mid$(tc$,b,1))-1
230 ifxp<0thennext:goto250
240 yp=val(mid$(tc$,b+1,1))-1:gosub
    270:fork=1tosp:next:next
250 x=x+8:ifx>152thenx=0:y=y+8:if
    y>200thenreturn
260 next:return
270 ch=int((2*(x+xp))/8):ro=int((
    y+yp)/8)
280 ln=(y+yp)and7:by=base+ro*320+
    8*ch+ln:poke1024+(ro*40)+ch,c
290 bi=7-((2*(x+xp))and7):pokeby,
    peek(by)or(2↑bi):return
300 rem letters
310 data 33,"4243444547"
320 data 34,"212213616253"
330 data 35,"2122232425262761626З
    6465666713233343"
340 data 35,"536373831525354555565
    7585"
350 data 36,"7262524232233344445464
    7566564636265152535455565 7"
360 data 37,"3122334281726354453 6
    2775667786"
370 data 38,"7766554433223141524З
    3425263747566574"
380 data 39,"515243"
390 data 40,"51424344454657"
400 data 41,"41525354555647"
410 data 42,"1424344454647422334 4
    55666253443526"
420 data 43,"4243444546243444546 4"
430 data 44,"555647"
440 data 45,"2434445464"
450 data 46,"47"
460 data 47,"81726354453627"
470 data 48,"413233343536475766 65
    64636251"
480 data 49,"42515253545556475767"
490 data 50,"2231415161726354453 6
    273747576777"
500 data 51,"2231415161727364545 4
    6475766757473726"
510 data 52,"5142332425354555657 5
    61626364656667"
520 data 53,"7161514131212223243 4
    44546475766757473727"
530 data 54,"7261514131222324252 6
    3747576777675645444З4"
540 data 55,"2221314151617172736 4
    555657"
550 data 56,"4434252637475767767 5
    6454232231415161727З"
560 data 57,"6151413122233444546 4
    73727374757667574737 26"
570 data 58,"4246"
580 data 59,"42454637"
590 data 60,"61524334455667"
600 data 61,"2333435363732535455 5
    6575"
610 data 62,"21324354453627"
620 data 63,"2231415161726354555 57"
630 data 64,"6353546474737261514 1
    323334353647576776"
640 data 193,"272625242332415162 7
    3747576773444546 4"
650 data 65,"2332425262737474757 677
    67574737263545556 5"
660 data 194,"212223242526273141 5
    16172736454443475766757473 7"
670 data 66,"2122232425262734445 4
    64757667574737"
680 data 195,"726151413122232425 2
    63747576776"
690 data 67,"7363534333242526374 7
    576777"
700 data 196,"212223242526273141 5
    161727374757667574737"
710 data 68,"7172737475767764544 4
    34252637475767"
720 data 197,"212223242526273141 5
    16171344454374757677 7"
730 data 69,"3545556574635343332 4
    25263747576777"
740 data 198,"212223242526273141 5
    16171344454"
750 data 70,"524131222324252627 34"
760 data 199,"737261514131222324 2
    526374757677675655 5"
770 data 71,"7363534333243545556 5
    74757667574737"
780 data 200,"212223242526273444 5
    464717273747576 77"
790 data 72,"2122232425262734445 4
    64757677"
800 data 201,"414243444546473151 3
    757"
810 data 73,"33434445464737575 41"
820 data 202,"515253545556473641 6
    1"
830 data 74,"53545556473651"
840 data 203,"212223242526276152 4
    334455667"
850 data 75,"2223242526274334455 6
    67"
860 data 204,"212223242526273747 5
    767"
870 data 76,"42525354555657476 7"
880 data 205,"111213141516172233 4
    453627172737475767 7"
890 data 77,"2425262734454647546 5
    6667"
```

```
 900 data 206,"2122232425262731425
     364616263656667"
 910 data 78,"23242526273343536465
     6667"
 920 data 207,"4131222324252637475
     76776757473726151"
 930 data 79,"43332425263747576776
     75746353"
 940 data 208,"2122232425262731415
     161727364544434"
 950 data 80,"22232425262732425262
     737465554535"
 960 data 209,"4131222324253646566
     6757473726151657687"
 970 data 81,"72737475767762524232
     232435455565"
 980 data 210,"2122232425262731415
     161727364544434455667"
 990 data 82,"22232425262732425262
     73"
1000 data 211,"7261514131222334445
     46475766757473726"
1010 data 83,"73635343332435455565
     766757473727"
1020 data 212,"5152535455565721314
     151617181"
1030 data 84,"424344445465767765464"
1040 data 213,"2122232425263747576
     7767574737271"
1050 data 85,"23242526374757677675
     7473"
1060 data 214,"2122232425364757667
     574737271"
1070 data 86,"23242536475766757473"
1080 data 215,"2122232425262736455
     56677767574737271"
1090 data 87,"23242526374656677675
     7473"
1100 data 216,"2122334455666761625
     344352627"
1110 data 88,"23344556676354453627"
1120 data 217,"2122334461625344454
     647"
1130 data 89,"22232435455562636465
     6657473727"
1140 data 218,"1121314151617162534
     4352617273747576777"
1150 data 90,"23334353635445362737
     475767"
1160 data 91,"61514142434445464757
     67"
1170 data 93,"31415152535455565747
     37"
1180 data 92,"72615141323334352617
     273747576776233343"
1190 data 94,"41424344454647233241
     5263"
1200 data 95,"14243444546474322341
     2536"
1210 data 192,"1222324252627282"
1220 data 0,"**"

*** EINDE LISTING ***

handwerk

regel 1     123
regel 2     135
regel 3     190
regel 4     143
regel 10    100
regel 20    159
regel 30    14
regel 40    20
regel 50    30
regel 60    196
regel 70    66
regel 80    39
regel 90    144
regel 100   225
regel 110   207
regel 120   39
regel 130   250
regel 140   113
regel 150   91
regel 160   54
regel 170   247
regel 180   34
regel 190   213
regel 200   77
regel 210   71
regel 220   221
regel 230   153
regel 240   209
regel 250   161
regel 260   74
regel 270   199
regel 280   162
regel 290   250
regel 300   178
regel 310   98
regel 320   188
regel 330   161
regel 340   77
regel 350   195
regel 360   132
regel 370   183
regel 380   147
regel 390   49
regel 400   53
regel 410   65
regel 420   106
regel 430   155
regel 440   100
regel 450   200
regel 460   61
regel 470   22
regel 480   125
regel 490   228
regel 500   187
regel 510   81
regel 520   125
regel 530   130
regel 540   160
regel 550   136
regel 560   247
regel 570   48
regel 580   4
regel 590   54
regel 600   64
regel 610   46
regel 620   103
regel 630   37
regel 640   233
regel 650   53
regel 660   237
regel 670   82
regel 680   170
regel 690   185
regel 700   184
regel 710   109
regel 720   176
regel 730   98
regel 740   149
regel 750   88
regel 760   89
regel 770   97
regel 780   222
regel 790   15
regel 800   254
regel 810   11
regel 820   154
regel 830   62
regel 840   47
regel 850   213
regel 860   1
regel 870   23
regel 880   60
regel 890   81
regel 900   109
regel 910   72
regel 920   229
regel 930   40
regel 940   98
regel 950   215
regel 960   85
regel 970   236
regel 980   156
regel 990   200
regel 1000   224
regel 1010   246
regel 1020   49
regel 1030   128
regel 1040   23
regel 1050   86
regel 1060   65
regel 1070   128
regel 1080   234
regel 1090   86
regel 1100   60
regel 1110   125
regel 1120   255
regel 1130   138
regel 1140   70
regel 1150   177
regel 1160   219
regel 1170   220
regel 1180   25
regel 1190   55
regel 1200   56
regel 1210   195
regel 1220   119
```

Klutskwijt 64

Dit spel drijft u tot wanhoop, want telkens, wanneer u denkt te weten hoe nu te manoevreren, verandert de joystickroutine en raakt u dus de kluts kwijt.

```
  1 rem klutskwijt / cbm-64
  2 rem door wim stadhouders
  3 rem terneuzen
  4 rem
100 goto770
110 rem lus
120 poke53278,0:poke53279,0
```

```
130 lw=lw+1:if lw=4then lw=0
140 lt=lt+1
150 if lw=0then poke2040,13
160 if lw=1then poke2040,14
170 if lw=2then poke2040,15
180 if lw=3then poke2040,253
190 i=peek(56320)
200 if i=a then ym=ym-3
210 if i=b then xm=xm+3
220 if i=c then ym=ym+3
230 if i=d then xm=xm-3
240 poke53248,xm:poke53249,ym
250 if xm>252then560
260 rem bonus
270 if lt=35then290
280 goto330
290 lt=0:bo=bo-1
300 if bo=-1then690
310 poke1779,bo+48
320 if bo=0then poke53280,15
330 if(peek(53279)and1)=1then690
340 if ti<wt then forp=1to60:next
    :goto110
350 rem happer
360 if h=1then430
370 vh=int(3*rnd(1))
380 if vh=0then xh=25:yh=ym:x1=sh
    :y1=0
390 if vh=1then yh=35:xh=xm:x1=0:
    y1=sh
400 if vh=2then yh=235:xh=xm:x1=0
    :y1=-1*sh
410 lh=0:h=1
420 poke53250,xh:poke53251,yh:poke
    53269,3:poke54276,129:poke542
    96,5
430 rem happer verplaatsen
440 lh=lh+1:if lh*sh>215then520
450 xh=xh+x1:yh=yh+y1:poke53250,x
    h:poke53251,yh
460 if lw=0then poke54273,15:poke
    2041,254
470 if lw=1then poke54273,18
480 if lw=2then poke54273,21:poke
    2041,255
490 if lw=3then poke54273,24
500 if peek(53278)=3then690
510 goto110
520 rem happer weg
530 h=0:poke54296,0:poke53269,1
540 ti$="000000"
550 goto110
560 rem einde ronde
570 sc=sc+r+bo
580 poke54276,33:poke54273,22:poke
    54272,96:poke53280,15:poke542
    96,15
590 print"[HOME][7xneer][BLAUW]"tab(
    33)sc
600 forp=1to400:next
610 poke54273,16:poke54272,195
620 forp=1to400:next
630 poke54273,28:poke54272,49
640 forp=1to400:next
650 poke54273,22:poke54272,96
660 forp=1to1000:next
670 forp=15to0step-.5:poke54296,p
    :next
680 poke53280,3:goto1460
690 rem dood
700 if sc>ts then ts=sc
710 poke54276,33:poke54273,28:poke
    54272,49:poke53280,2:poke5429
    6,15
720 forp=1to500:next
730 poke54273,14:poke54272,24:poke
    53280,1
740 forp=1to1000:next
750 forp=15to0step-.5:poke54296,p
    :next
760 poke53280,1:goto1290
770 print"[CLR-HOME]":poke53281,1
    :poke53280,1
780 poke53269,0
790 forp=54272to54296:pokep,0:next
800 poke53279,0
810 poke54277,64:poke54278,128
820 poke54276,129:poke54296,15
830 forp=55336to55375:pokep,p-553
    36:next
840 forp=1064to1088:pokep-1,32:poke
    p,25:poke54273,p-1062
850 forq=1to30:nextq:nextp
860 forp=1064to1086:pokep-1,32:poke
    p,15:poke54273,p-1062
870 forq=1to30:nextq:nextp
880 forp=1064to1084:pokep-1,32:poke
    p,10:poke54273,p-1062
890 forq=1to30:nextq:nextp
900 forp=1064to1080:pokep-1,32:poke
    p,4:poke54273,p-1062
910 forq=1to30:nextq:nextp
920 forp=1064to1078:pokep-1,32:poke
    p,1:poke54273,p-1062
930 forq=1to30:nextq:nextp
940 forp=1064to1076:pokep-1,32:poke
    p,2:poke54273,p-1062
950 forq=1to30:nextq:nextp
960 poke54296,0
970 print"[3xneer][BLAUW]"tab(14)
    "s t i c k"
980 print"[PURPER][3xneer]"tab(12
    )"▒▒▒▒▒▒▒▒▒▒▒▒▒"
990 forp=1to11:printtab(12)"▒[11xspat
    ie]▒":next
1000 printtab(12)"▒▒▒▒▒▒▒▒▒▒▒▒▒"
1010 poke1442,14:poke55714,12
1020 poke1402,66:poke55674,3
1030 poke1802,66:poke56074,3
1040 poke1762,19:poke56034,12
1050 poke1597,67:poke55869,3
1060 poke1598,23:poke55870,12
1070 poke1606,5 :poke55878,12
1080 poke1607,67:poke55879,3
1090 forp=832to894:reada:pokep,a:next
1100 poke54276,17:poke54273,16:poke
     54272,195:poke54296,15
1110 poke2040,13:poke53277,1:poke5
     3271,1:poke53287,14:poke53248
     ,156
1120 poke53249,150:poke53269,1
1130 forp=896to958:reada:pokep,a:next
1140 poke1442,5:poke1762,23:poke15
     98,14:poke1606,19:poke2040,14
     :poke54273,18
1150 forp=960to1022:reada:pokep,a:
     next
1160 poke1442,19:poke1762,5:poke15
     98,23:poke1606,14:poke2040,15
1170 poke54273,21:poke54272,31
1180 forp=16192to16254:reada:pokep
     ,a:next
1190 poke1442,23:poke1762,14:poke1
     598,5:poke1606,19:poke2040,25
     3
1200 poke54273,22:poke54272,96
1210 forp=1to400:next
1220 poke54273,25:poke54272,30:poke
     2040,13
1230 poke1442,63:poke1762,63:poke1
     598,63:poke1606,63
```

```
1240 forp=1to1000:next
1250 poke54296,0
1260 forp=16256to16318:reada:pokep
     ,a:next
1270 forp=16320to16382:reada:pokep
     ,a:next
1280 forp=1to2000:next
1290 poke53269,0:poke53277,0:poke5
     3271,0:print"[CLR-HOME]";
1300 poke53279,0
1310 poke53287,6:poke53288,9:u=129
     4
1320 rem schermtekening
1330 print"[PURPER]▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
     ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
1340 forp=1to23:print"▒"tab(30)"▒"
     :next
1350 print"[PURPER]▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
     ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒[HOME]"
1360 printtab(33)"[ROOD]ronde"
1370 print"[2xneer][GROEN]"tab(14)
     "▚▚"tab(25)"▚▚"
1380 printtab(33)"[ROOD]score"
1390 print"[2xneer][GROEN]"tab(8)"
     ▚▚▚"tab(19)"▚▚▚"
1400 printtab(34)"[ROOD]top"
1410 printtab(33)"score"
1420 print"[neer][GROEN]"tab(14)"▚
     ▚"tab(25)"▚▚"tab(33)"[BLAUW]"
     ts
1430 print"[3xneer][GROEN]"tab(8)"
     ▚▚▚"tab(19)"▚▚▚"tab(33)"[ROOD]
     bonus"
1440 print"[GROEN][3xneer]"tab(14)
     "▚▚"tab(25)"▚▚"
1450 r=0:sc=0
1460 rem start ronde
1470 poke53269,0:poke53280,1
1480 rem uitgang
1490 h=0:poke53278,0:poke53279,0
1500 poke u+54272,4:poke u,102:poke
      u+54312,4:poke u+40,102
1510 poke u+54352,4:poke u+80,102
1520 ub=int(19*rnd(1))
1530 u=1134+(ub*40)
1540 poke u,32:poke u+40,32:poke u
     +80,32
1550 rem stootblok
1560 hs=int(19*rnd(1))
1570 vs=int(23*rnd(1))
1580 sb=1072+hs+(vs*40)
1590 if peek(sb)=32then1610
1600 goto1550
1610 pokesb,86:pokesb+54272,8
1620 r=r+1:print"[HOME][3xneer][BLAUW]"
     tab(34)r
1630 sh=int((r*.4)+.6)
1640 wt=500-(r*20)
1650 xm=54:ym=140:poke53248,xm:poke
     53249,ym:poke2040,13:poke5326
     9,1
1660 rem definitie
1670 a=int(4*rnd(1))
1680 if a=0then a=126
1690 if a=1then a=119
1700 if a=2then a=125
1710 if a=3then a=123
1720 b=int(4*rnd(1))
1730 if b=0then b=126
1740 if b=1then b=119
1750 if b=2then b=125
1760 if b=3then b=123
1770 if b=athen1720
1780 c=int(4*rnd(1))
1790 if c=0then c=126
1800 if c=1then c=119
1810 if c=2then c=125
1820 if c=3then c=123
1830 if c=athen1780
1840 if c=bthen1780
1850 d=int(4*rnd(1))
1860 if d=0then d=126
1870 if d=1then d=119
1880 if d=2then d=125
1890 if d=3then d=123
1900 if d=athen1850
1910 if d=bthen1850
1920 if d=cthen1850
1930 ti$="000000":bo=9:lt=0:poke17
     79,57:poke56051,10
1940 goto110
1950 rem fase 1
1960 data255,255,0,255,255,0,193,1
     31,0,193,131,0,193,131,0,193,
     131,0,193,131,0
1970 data193,131,0,193,131,0,193,1
     31,0,193,131,0,193,131,0,193,
     131,0,193,131,0
1980 data255,255,0,255,255,0,0,0,0
     ,0,0,0,0,0,0,0,0,0,0,0,0
1990 rem fase 2
2000 data255,255,0,255,255,0,192,1
     5,0,192,15,0,192,51,0,192,51,
     0,192,195,0
2010 data192,195,0,195,3,0,195,3,0
     ,204,3,0,204,3,0,240,3,0,240,
     3,0
2020 data255,255,0,255,255,0,0,0,0
     ,0,0,0,0,0,0,0,0,0,0,0,0
2030 rem fase 3
2040 data255,255,0,255,255,0,192,3
     ,0,192,3,0,192,3,0,192,3,0,19
     2,3,0
2050 data255,255,0,255,255,0,192,3
     ,0,192,3,0,192,3,0,192,3,0,19
     2,3,0
2060 data255,255,0,255,255,0,0,0,0
     ,0,0,0,0,0,0,0,0,0,0,0,0
2070 rem fase 4
2080 data255,255,0,255,255,0,240,3
     ,0,240,3,0,204,3,0,204,3,0,19
     5,3,0
2090 data195,3,0,192,195,0,192,195
     ,0,192,51,0,192,51,0,192,15,0
     ,192,15,0
2100 data255,255,0,255,255,0,0,0,0
     ,0,0,0,0,0,0,0,0,0,0,0,0
2110 rem happer hoog
2120 data32,0,4,48,0,12,24,60,24,1
     2,126,48,6,255,96,3,189,192,3
     ,24,192
2130 data7,189,224,15,255,240,31,2
     31,248,63,231,252,127,231,254
     ,121,255,158
2140 data124,255,62,62,126,124,31,
     0,248,15,129,240,7,255,224,3,
     255,192
2150 data1,255,128,0,0,0
2160 rem plat
2170 data0,0,0,0,0,0,0,60,0,0,126,
     0,255,255,255,255,189,255,3,2
     4,192
2180 data7,189,224,15,255,240,31,2
     31,248,63,231,252,127,231,254
     ,127,255,254
2190 data127,255,254,63,255,252,31
     ,129,248,15,255,240,7,255,224
     ,3,255,192
2200 data1,255,128,0,0,0

*** EINDE LISTING ***
```

```
klutskwijt

regel 1    67
regel 2    246
regel 3    79
regel 4    143
regel 100   39
regel 110   131
regel 120   51
regel 130   77
regel 140   205
regel 150   164
regel 160   166
regel 170   168
regel 180   221
regel 190   14
regel 200   74
regel 210   72
regel 220   75
regel 230   75
regel 240   24
regel 250   188
regel 260   22
regel 270   135
regel 280   31
regel 290   108
regel 300   240
regel 310   66
regel 320   208
regel 330   2
regel 340   252
regel 350   79
regel 360   244
regel 370   114
regel 380   84
regel 390   85
regel 400   16
regel 410   219
regel 420   22
regel 430   148
regel 440   220
regel 450   193
regel 460   68
regel 470   233
regel 480   68
regel 490   232
regel 500   210
regel 510   27
regel 520   50
regel 530   152
regel 540   215
regel 550   27
regel 560   108
regel 570   21
regel 580   217
regel 590   78
regel 600   168
regel 610   207
regel 620   168
regel 630   160
regel 640   168
regel 650   156
regel 660   213
regel 670   135
regel 680   134
regel 690   181
regel 700   15
regel 710   169
regel 720   169
regel 730   196
regel 740   213
regel 750   135
regel 760   133
regel 770   209
regel 780   252
regel 790   110
regel 800   253
regel 810   216
regel 820   212
regel 830   64
regel 840   203
regel 850   213
regel 860   200
regel 870   213
regel 880   193
regel 890   213
regel 900   144
regel 910   213
regel 920   148
regel 930   213
regel 940   147
regel 950   213
regel 960   253
regel 970   34
regel 980   141
regel 990   8
regel 1000   122
regel 1010   89
regel 1020   49
regel 1030   48
regel 1040   95
regel 1050   71
regel 1060   104
regel 1070   54
regel 1080   64
regel 1090   181
regel 1100   169
regel 1110   93
regel 1120   151
regel 1130   192
regel 1140   245
regel 1150   215
regel 1160   139
regel 1170   144
regel 1180   120
regel 1190   191
regel 1200   156
regel 1210   168
regel 1220   186
regel 1230   157
regel 1240   213
regel 1250   253
regel 1260   122
regel 1270   115
regel 1280   214
regel 1290   69
regel 1300   253
regel 1310   83
regel 1320   166
regel 1330   147
regel 1340   79
regel 1350   166
regel 1360   163
regel 1370   5
regel 1380   167
regel 1390   89
regel 1400   31
regel 1410   139
regel 1420   48
regel 1430   131
regel 1440   22
regel 1450   230
regel 1460   149
regel 1470   44
regel 1480   158
regel 1490   151
regel 1500   1
regel 1510   240
regel 1520   162
regel 1530   114
regel 1540   11
regel 1550   80
regel 1560   166
regel 1570   175
regel 1580   10
regel 1590   185
regel 1600   84
regel 1610   62
regel 1620   50
regel 1630   18
regel 1640   78
regel 1650   151
regel 1660   32
regel 1670   22
regel 1680   225
regel 1690   228
regel 1700   226
regel 1710   225
regel 1720   23
regel 1730   227
regel 1740   230
regel 1750   228
regel 1760   227
regel 1770   49
regel 1780   24
regel 1790   229
regel 1800   232
regel 1810   230
regel 1820   229
regel 1830   56
regel 1840   57
regel 1850   25
regel 1860   231
regel 1870   234
regel 1880   232
regel 1890   231
regel 1900   55
regel 1910   56
regel 1920   57
regel 1930   233
regel 1940   27
regel 1950   223
regel 1960   173
regel 1970   161
regel 1980   147
regel 1990   224
regel 2000   246
regel 2010   70
regel 2020   147
regel 2030   225
regel 2040   190
regel 2050   190
regel 2060   147
regel 2070   226
regel 2080   169
regel 2090   147
regel 2100   147
regel 2110   124
regel 2120   60
regel 2130   65
regel 2140   66
regel 2150   87
regel 2160   192
regel 2170   181
regel 2180   68
regel 2190   234
regel 2200   87

ready.
```

Rattlesnake

Ga er maar weer achteraan, met dit achtervolgingsspel, waarbij het ontwijken van obstakels veel van de stuurcapaciteiten vraagt.




```
  1 rem rattlesnake / cbm-64
  2 rem door peter abe
  3 rem schriek / 015 23 40 54(be
    lgie)
  4 rem
 10 h=-13:poke 53280,0:poke 53281
    ,0:poke 646,4
 20 poke 56334,peek(56334) and 25
    4
 30 s=54272:poke s+24,15:poke s,0
    :poke s+1,25:poke s+5,128:poke
     s+6,0
 40 p=-13:print"[CLR-HOME]":fori=
    1to25:print:next
 50 poke 1444+x,98:p=p+1
 60 j=peek(56320):x=x+(j=123)-(j=
    119)
 70 print:a=1904+rnd(1)*40:for i=
    a to a+p/150:pokei,102:next
 80 poke 54276,33:poke 54276,130
 90 if peek(1444+x)<>102 then 50
100 if p>h then h=p
```

```
110 poke s+6,0:poke s+5,12:poke s
    +4,20: poke s+4,129:for i=1 to
     2500:next
120 print"[HOME]";:print"[BLAUW][10xs
    patie]O‾‾‾‾‾‾‾‾P[11xspatie]O‾‾
    ‾‾‾‾‾p"
130 print"[GEEL] hi score:[BLAUW]
    | [7xspatie] |[4xspatie][GEEL]
    score:[BLAUW]| [7xspatie] |";:print
    "[8xlinks][WIT]"p
140 print"[op][11xrechts]";h:print
    "[BLAUW][10xspatie]L▁▁▁▁▁▁▁▁▲[11xs
    patie]L▁▁▁▁▁▁▁▁▲"
150 poke s+6,240:poke s+5,0:poke
    s+4,0: poke s+4,17
160 for i=20 to 6 step-1:for j=i
    to 50-i:poke s+1,j:nextj,i
170 print:print:print"[CYAAN][12xspat
    ie][RVS-aan][2xspatie]new gam
    e(y/n)[2xspatie][RVS-uit][PURPER]
    [3xspatie]"
180 poke 56334,peek(56334) or 1
190 poke s+1,3:poke s+4,0
200 poke s+4,33:for i=1 to 50:next
210 poke s+4,0:for i=1 to 200:next
220 geta$:if a$="" then 200
230 if a$="y" then 20

*** EINDE LISTING ***
```

```
rattlesnake

regel 1     101            regel 120    161
regel 2     11             regel 130    159
regel 3     136            regel 140    4
regel 4     143            regel 150    223
regel 10     57            regel 160    42
regel 20     42            regel 170    141
regel 30     11            regel 180    193
regel 40     60            regel 190    130
regel 50     106           regel 200    6
regel 60     203           regel 210    253
regel 70     57            regel 220    95
regel 80     202           regel 230    72
regel 90     112
regel 100     197
regel 110     32           ready.
```

S.I.T.

Vanwege de gevolgde programmeermethode is het van belang dat u deze listing uiterst nauwkeurig intikt. (let ook op het aantal spaties etc.) Bij het runnen moet u anderhalve minuut geduld hebben. Nadat u het programma dan gerund heeft gebeurt er ogenschijnlijk niets. Uw computer bezit nu echter een aantal extra mogelijkheden. Om deze te leren kennen is er maar een manier en dat is de DEMO van SIT te bestuderen. Daar dit een bijzonder lange listing is raden wij u aan om van te voren even te overwegen of u er niet verstandiger aandoet om de INFOLIST cassette te bestellen. In praktijk blijkt nu eenmaal dat zelfs moeders niet voor een dergelijke hoeveelheid aanslagen te porren zijn.

```
1 rem s.i.t.  64
2 rem door raoul teeuwen
3 rem gorichem / 01830-22171
4 rem
10 poke55,255:poke56,154:clr:rem
   sit door r.teeuwen - 1985
20 fora=0to31:reada$:forb=1tolen(a
   $)step2:b$=mid$(a$,b,2):c$=left$
   (b$,1):gosub50
30 c=c+16*e:c$=right$(b$,1):gosub5
   0:c=c+e:poke39680+d,c:c=0:d=d+1
40 next:next:sys40640:end
50 e=val(c$):ifasc(c$)<58thenreturn
60 e=(asc(c$)-55):return
70 data78a9418d1403a99b8d1503a93b8
   d12d0ad11d0297f8d11d0a9818d1ad0
   586000013bf078
80 dataa93c8d219ba9f08d229ba9418d0
   29ba99b8d079ba90f8d599b584c389c
   ad19d08d19d029
90 data01d007ad0ddc584c31eaad12d0c
   d229bb00fad209b8d20d08d21d0ad22
   9b4c7d9ba9008d
100 data20d08d21d0ac860220839b8c860
    2ad219b8d12d04cbcfea228a9019dff
    d79dbfdbcad0f7
110 dataa90520d2ffa228a9209dff039d9
    7079dbf07cad0f4a200a9a09d00049d
    c007e8e8e8e02a
120 datad0f36078a92d8d219ba9fe8d229
    ba9418d029ba99b8d079ba97b8d599b
    584ce49bad1f9b
130 data8d20d08d21d0ad219b4c7d9ba92
    0a2009d00049dc007e8e028d0f54c38
    9c78a9038d1403
140 dataa99c8d15035860ee369cad369cc
    91ff0034c31eaa9008d369cad379cd0
    0eee379cad11d0
150 data09408d11d04cbcfece379cad11d
    029bf8d11d04cbcfe1601ad219b8d12
    d0ad11d029bf8d
160 data11d04c009b78a9f08d1ad0584cf
    69bc61f78a9318d1403a9ea8d1503a9
    f08d1ad058ee53
170 data9cad539c29028d20d08d21d0ac5
    49cc8d0fda5c6f0e860ea78a9ea8d71
    9c8d729c8d739c
180 data586078a98d8d719ca9218d729ca
    9d08d739c5860e9f2fd78a9ca8d1403
    a99c8d1503ad11
190 datad0297f8d11d0a9818d1ad0a9058
    de49c20839ca9e98d12d05860ad19d0
    8d19d02901d007
200 dataad0ddc584c31eaad12d0cda19cb
    01420059dada29c29078d16d0ada19c
    8d12d04cd19da9
210 datac88d16d0ada09c8d12d04c81eae
    ea29c607820a39ca9148de49c5860ce
    a29c6078ad11d0
220 data29bf8d11d0a9c88d16d0a9318d1
    403a9ea8d1503a9f08d1ad0586078a9
    648d1403a99d8d
230 data1503ad11d0297f8d11d0a9818d1
    ad0a9018d12d0a22fbdffcf9d9f9f9d
    cf9fcad0f45860
240 dataad19d08d19d02901d007ad0ddc5
    84c31eaad12d0c978b026a219bd9f9f
    9dffcfcad0f7a2
250 data15bdb99f9d19d0cad0f7a208bd8
    f9f9df707cad0f7a9784cc59da219bd
    cf9f9dffcfcad0
260 dataf7a215bde99f9d19d0cad0f7a20
    8bd979f9df707cad0f7a9018d12d04c
    81ea20379d4c38
270 data9cade49cc914f01aada29c2907c
    907f0034c81eaa227bd97079d9807ca
    d0f74c81eaada2
280 data9c2907f0034c81eaa200bd99079
    d9807e8e027d0f54c81ea78a9008d12
    d4a9128d0fd4a2
```

```
290 data208a49ff8d01d4ec12d0d0fbad1
    1d02980d0f4a9178d11d0a9088d16d0
    a9008d01d4ad12
300 datad0d0fbad11d02980d0f4a9578d1
    1d0a9188d16d0e8e0ffd0c4a9178d11
    d0a9088d16d0a9
310 data1b8d11d0586078a9008dbf9ea92
    08d189ea9e88d509ea9ff8d529e200c
    9ea98d8d1403a9
320 data9e8d150320989e5860adbf9ed00
    c20989e4c31eaa9008d11d06078a9ff
    8d189ea9ca8d50
330 data9ea9208d529ea9318d1403a9ea8
    d150358200c9e4c31ea00a99a8538a5
    9dd0034cda9ea2
340 data00bd509f499820d2ffe8e020d0f
    378a9e78d0803a99e8d090358602073
    00c92ef0062079
350 data004ce7a720730038e930c90a900
    62079004cee9e0a0aa8b9289f8d209f
    b9299f8d219fb9
360 data2a9f8d239fb92b9f8d249f20aaa
    a20aaaa6c080318 9d669e239be69eb7
    9be69e499ce69e
370 data809c559c8e9c559ca39ce69e099
    de69e189de69e379de69e8989b8b8b8
    b8b8b8b8b8b8b8
380 data8acbd1ccb8d6d7cfb8d9dbccd1c
    ed9ccdddc0c95

*** EINDE LISTING ***
```

s.i.t.

regel	1	115		
regel	2	99		
regel	3	50		
regel	4	143		
regel	10	47		
regel	20	163		
regel	30	221		
regel	40	206		
regel	50	12		
regel	60	163		
regel	70	89		
regel	80	235		
regel	90	229		
regel	100	12		
regel	110	210		
regel	120	33		
regel	130	120		
regel	140	241		
regel	150	6		
regel	160	190		
regel	170	28		
regel	180	207		
regel	190	145		
regel	200	3		
regel	210	234		
regel	220	170		
regel	230	31		
regel	240	49		
regel	250	183		
regel	260	17		
regel	270	11		
regel	280	161		
regel	290	167		
regel	300	209		
regel	310	233		
regel	320	215		
regel	330	169		
regel	340	92		
regel	350	141		
regel	360	44		
regel	370	89		
regel	380	103		

ready.

```
   LISTING programma : demo sit

1 fl=0:sys40640
2 print"[CLR-HOME][6xneer][ZWART]"
3 print"[6xspatie]■[RVS-aan][7xspatie
```

```
   ][RVS-uit][5xspatie][RVS-aan]▪ ▪
   [RVS-uit] [RVS-aan][9xspatie][RVS-u
   it][3xspatie]▪▪"
 4 print"[6xspatie][RVS-aan] [RVS-uit]
   ▪[5xspatie]▪▪[4xspatie][RVS-aan]
   ▪ ▪[RVS-uit][4xspatie]▪[RVS-aan]
    [RVS-uit]▪[6xspatie][RVS-aan]▮
   [RVS-uit]▮"
 5 print"[6xspatie][RVS-aan] [RVS-uit]
   [12xspatie]▪▪[5xspatie][RVS-aan]
    [RVS-uit][7xspatie][RVS-aan]▮[RVS-
   uit]▮"
 6 print"[6xspatie][RVS-aan] [RVS-uit]
   [12xspatie][RVS-aan][2xspatie][RVS-
   uit][5xspatie][RVS-aan] [RVS-uit]
   [7xspatie][RVS-aan]| [RVS-uit]| "
 7 print"[6xspatie][RVS-aan] [RVS-uit]
   ▪[11xspatie][RVS-aan][2xspatie]
   [RVS-uit][5xspatie][RVS-aan] [RVS-u
   it][7xspatie][RVS-aan] [RVS-uit]"
 8 print"[6xspatie]▪[RVS-aan][6xspatie
   ][RVS-uit]▪[5xspatie][RVS-aan][2xsp
   atie][RVS-uit][5xspatie][RVS-aan]
    [RVS-uit][6xspatie] |[RVS-aan] |
   [RVS-uit]"
 9 print"[12xspatie]▪[RVS-aan] [RVS-ui
   t][5xspatie][RVS-aan][2xspatie]
   [RVS-uit][5xspatie][RVS-aan] [RVS-u
   it][6xspatie]▮[RVS-aan]▮[RVS-uit]"
10 print"[13xspatie][RVS-aan] [RVS-uit
   ][5xspatie][RVS-aan][2xspatie][RVS-
   uit][5xspatie][RVS-aan] [RVS-uit]
   [6xspatie][RVS-aan]▮[RVS-uit]▮"
11 print"[13xspatie][RVS-aan] [RVS-uit
   ][5xspatie][RVS-aan][2xspatie][RVS-
   uit][5xspatie][RVS-aan] [RVS-uit]
   [6xspatie][RVS-aan]| [RVS-uit]| "
12 print"[5xspatie]▪▪[5xspatie]▪[RVS-a
   an] [RVS-uit] [RVS-aan]▪ ▪[RVS-uit]
   [2xspatie][RVS-aan][2xspatie][RVS-u
   it] [RVS-aan]▪ ▪[RVS-uit][2xspatie]
   [RVS-aan] [RVS-uit]"
13 print"[5xspatie]▪[RVS-aan][7xspatie
   ][RVS-uit]▪ [RVS-aan]▪ ▪[RVS-uit]
    ▪[RVS-aan][2xspatie][RVS-uit]▪
   [RVS-aan]▪ ▪[RVS-uit][2xspatie][RVS-
   aan] [RVS-uit][5xspatie] |[RVS-aan]
    |[RVS-uit]"
14 .1:fora=0to6000:next
15 print"[HOME][4xneer][ROOD]"tab(
   11)"c-info presenteert:"
16 print"[15xneer][11xspatie][BLAUW]
   [RVS-aan]super interrupt tool[RVS-u
   it]"
17 fora=0to5000:next
18 .2:poke39711,7:poke39712,14
30 fora=0to20000:next:.8
31 poke53280,0:poke53281,0:print"[CLR-
   HOME][GEEL][6xneer][RVS-aan][34xspa
   tie][RVS-uit]";
32 print"[RVS-aan][11xspatie]dit i
   s een kleine demonstratie[23xspatie
   ]";
33 print"[RVS-aan][30xspatie]van h
   et '[RVS-uit]flash[RVS-aan]' co
   mmando.[10xspatie]";
34 print"[RVS-aan][41xspatie]er zi
   jn geen timings-probl";
35 print"[RVS-aan]emen ![56xspatie]
   [RVS-uit]";
36 print"[RVS-aan]flashing is simp
   le[50xspatie][RVS-uit]"
37 .3:fora=0to9000:next
38 .8:poke53280,0:poke53281,6
39 print"[CLR-HOME][WIT][5xspatie]
   s.i.t. (super interrupt tool)"
40 print"[7xspatie]*** een kleine
   uitleg ***"
41 print"[neer][ZWART] hoewel sit
   in direct mode te gebruiken"
42 print" is, is hij gemaakt voor
   mensen die wel"
43 print" in basic kunnen programe
   ren, maar waar"
44 print" machinetaal nog hokuspok
   us voor is."
45 print" s.i.t. kan een basicprog
   ramma voorzien"
46 print" van leuke effecten, die
   het geheel wat"
47 print" opfleuren."
48 print" de commando's die hiervo
   or voorhande"
49 print" zijn, zijn de volgende:"
50 print"[neer] * .1 - de filminte
   rrupt"
51 print" * .2 - de nieuwe border
   interrupt"
52 print" * .3 - de flash interrup
   t"
53 print" * .4 - de wacht-scroll i
   n de border"
54 print" * .5 - zie .4, over het
   hele scherm"
55 print" * .6 - smooth scroll naa
   r rechts"
56 print" * .7 - smooth scroll naa
   r linkss"
57 print" * .8 - zet alles op stan
   daard waarden"
58 print" * .9 - 16 sprites interr
   upt"
59 print" * .0 - screen magic"
60 print"[12xspatie][RVS-aan] druk
    een toets [RVS-uit]"
61 .4:print"[CLR-HOME][neer][WIT][5xsp
   atie]s.i.t. (super interupt to
   ol)"
62 print"[7xspatie]*** een kleine
   uitleg ***"
63 print"[neer] om s.i.t compatibl
   e te maken met andere";
64 print" toolkits, zoals snellade
   rs, is hij net"
65 print" onder de basic-rom gepla
   atst."
66 print" s.i.t neemt 1,25 kbyte v
   an het geheugen";
67 print" af, en verplaatst de top
    of basic."
68 print" s.i.t staat van $9b00 to
   t $9fff."
69 print"[neer] er is geprobeerd s
   .i.t zo gebruikers-"
70 print" vriendelijk mogelijk te
   maken, maar"
71 print" wegens de complexiteit v
   an s.i.t, zal"
72 print" het nodig zijn om met de
   ze tool te"
73 print" oefenen.na enige tijd za
   l dan deze"
74 print" tool beslist een veel ge
   bruikt"
75 print" programmatje worden."
76 print"[neer] deze demo laat de
   mogelijkheden zien,"
77 print" maar met wat creativitei
   t is er nog"
78 print" meer uit te halen!"
79 print"[12xspatie][RVS-aan] druk
    een toets [RVS-uit]":poke198,0:.5
```

```
80 printchr$(14)
81 poke53280,0:poke53281,0:print"[CLR-
   HOME]"
82 print"[2xneer][2xspatie]Dit is
   een demo van het .6 en .7"
83 print"[neer][2xspatie]commando.
   Het toont de smooth-scroll"
84 print"[neer][2xspatie]naar link
   s en naar rechts, van een"
85 print"[neer][2xspatie]deel van
   het scherm."
86 print"[2xneer][2xspatie]Zoals t
   e zien is bij proffesionele"
87 print"[neer][2xspatie]intro's,
   geeft zo'n interrupt een"
88 print"[neer][2xspatie]leuk effe
   ct !"
89 print"[3xneer][2xspatie]WEER[shift-
   SPATIE]EEN[shift-SPATIE]COMMAND
   O[shift-SPATIE]VAN[shift-SPATIE]
   S.I.T !"
90 fora=0to4000:next
91 fora=56176to56295:pokea,7:next:
   rem   kleur poken,alleen oudere
    64's !
92 a$=" een van de inzendingen voo
   r de c-info miniatuur wedstrijd
    ......."
93 a$=a$+" s.i.t door r.teeuwen !!
   !!!!! "
94 fora=1944to1983:pokea,63:next
95 .6:fora=1tolen(a$):poke1944,asc
   (mid$(a$,a,1)):forb=0to121:next
   :next
96 fora=0to6000:next:.8:fora=1944to
   1983:pokea,33:next:.7
97 fora=1tolen(a$):poke1983,asc(mid$
   (a$,a,1)):forb=0to120:next:next
98 fora=0to6000:next
99 .8:print"[CLR-HOME]":poke53280,
   0:poke53281,0:printchr$(142):.9
   :iffl=1then140
100 clr:print"[10xneer][3xspatie]ik
    lees nu 17 sprites in ......."
101 print"[2xneer][2xspatie]even ge
    duld a.u.b.[2xspatie](70 second
    en !)"
102 rem 17 sprites  1.10 min conver
    ting!
103 restore:fora=1to31
104 reada$:forb=1tolen(a$)step2:b$=
    mid$(a$,b,2):c$=left$(b$,1):gosub
    107
105 c=c+16*e:c$=right$(b$,1):gosub1
    07:c=c+e:poke12288+d,c:c=0:d=d+
    1
106 next:next:goto140
107 e=val(c$):ifasc(c$)<58thenreturn
108 e=(asc(c$)-55):return
109 data3ffc007ffc00fffc00fffc00fc0
    7fef007fcf007f8f007f0f007e0f007
    c0f00000f007c0
110 dataf007e0f007f0f007f8f007fcfc0
    7fefffc00fffc007ffc003ffc000000
    00001ffff83fff
111 datafc7ffffe7ffffe7f01fe7e007e7
    e007e7e007e7e007e7e007e7e007e7e
    007e7e007e7e00
112 data7e7f00fe7ffffe7ffffe3ffffc1
    ffff80000000e03c03e07c0ff0de1b
    d99b33cf19e3c6
113 data18c3c01803c01803c01803c0180
    3c01803c01803c01803c01803c01803
    c01803c01803c0
114 data1803c01803c01803c0180300000
    00000000ffffc0fffff0fffff81fff
    fc1f03fc1f01fe
115 data1f00fe1f00fe1f00fe1f00fe1f0
    0fe1f00fe1f01fe1f03fc1ffffcffff
    f8fffff0ffffc0
116 data00000000000000fff80ffffe0f
    ffff0ffff80f81f80f80fc0f80fc0f
    80fc0f80fc0f81
117 dataf80fff80fff00fffe00fff800
    ff0000ff8000f7c000f3e000f1f000f
    0f80003fffff7f
118 dataffffffffffffffffffffffc000
    0fc0000fe0000fffe00ffff00ffff00
    fffe00fe0000fc
119 data0000fc0000fe0000ffffffffff
    fffffff7ffff3ffff003ffffc7fff
    feffffffffffff
120 dataffffff7ffffe3ffffc007e00003
    c00003c00003c00003c00003c00003c
    00007e003ffffc
121 data7fffefffffffffffffff7ffffe3ff
    ffc00fe01f8ff01f8ff01f8ff81f8ff
    81f8ffc1f8ffc1
122 dataf8fbe1f8fbe1f8f9f1f8f9f1f8f
    8f9f8f8f9f8f87df8f87df8f83ff8f8
    3ff8f81ff8f81f
123 dataf8f80ff8f80ff8003ffff07ffff
    8ffffcffffcffff8ffff0fe0000
    fc0000fc0000fc
124 data0000fc0000ffe000fff000fff00
    0ffe000fc0000fc0000fc0000fc0000
    fc000078000000
125 data000000000f80001fc0001fc000
    3f80003f80007f00007f0000fe0000f
    e0001fc0001fc0
126 data003f80003f80001f0000000000
    00000fc0001fc0003f80001f000000
    00003f00007f80
127 data007f80003fc0003fc0001fe0001
    fe0000ff0000ff00007f80007f80003
    fc0003fc0001f8
128 data0000000000000000003f00003f800
    01fc0000fc000000000000000000000
    0000000000000
129 data00003ffc007ffe00fffe00fffc0
    0f807fef007fcf007f0f00700f007f0
    f007fcf807feff
130 datafc00fffe007ffe003ffc0000000
    000000000000000000000000000000
    0000000000000
131 data0000001ffff83ffffc7ffffe7ff
    ffe7c003e78001e78001e7c003e7fff
    fe7ffffe3ffffc
132 data1ffff8000000000000000000000
    000000000000000000000000000000
    0000000000c03c
133 data03e07e07f0db0fd9999bcf18f3c
    61863c01803c01803c01803c01803c0
    1803000000000
134 data000000000000000000000000000
    00000000000000ffff0ffffcffffe
    fffff0fc07f0f
135 data803f0f803f0f803f0fc07fffff
    ffffffeffffcffff000000000000000
    0000000000000
136 data0000000000000000000000000000
    000fffe0fffff0ff81f8ff00fc1f81
    fc1ffff81ffff0
137 data1fe0001ff8001ffe001e7f001e1
    f800000000000000000000000000000
    0000000000000
138 data007ffff8ffffcffffcffff8f
    00000ffe000fff000ffe000f00000f0
    0000ffffcffff
139 datafeffffe7ffffc
140 print"[CLR-HOME][8xneer][GEEL]▬
    ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬
```

```
    ▬▬▬▬▬▬▬▬"
141 print"[11xneer]▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬
    ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬"
142 print"[9xop] 16 sprites tegelij
    k op het scherm !!!!"
143 print"[3xspatie]met meer dan 8
    verschillende ....."
144 print" oplopend tot 16 verschil
    lende sprites!"
145 v1=40864:v2=40912:p1=40848:p2=4
    0856
146 fori=0to7:pokev1+2*i+1,0:pokev2
    +2*i+1,255:next
147 fori=0to7:pokev1+2*i,i*24+80:next
148 fori=0to5:pokev1+2*i,i*24+80:poke
    v2+2*i,i*24+110:next
149 pokev1+21,255:fora=0to7:pokev1+
    39+a,1:next
150 pokev2+21,63:fora=0to5:pokev2+3
    9+a,1:next
151 pokep1,192:pokep1+1,193:pokep1+
    2,194:pokep1+3,193:pokep1+4,195
152 pokep1+5,193:pokep1+6,196:pokep
    1+7,197
153 pokep2,202:pokep2+1,198:pokep2+
    2,199:pokep2+3,200:pokep2+4,193
154 pokep2+5,201
155 fora=0to75:forb=0to7:pokev1+b*2
    +1,a:next:next
156 fora=255to200step-1:forb=0to7:poke
    v2+b*2+1,a:next:next
157 poke198,0
158 print"[HOME][23xneer][6xspatie]
    [RVS-aan]druk een toets voor ve
    rder[RVS-uit]"
159 fora=75to98:forb=0to7:pokev1+2*
    b+1,a:next:c=peek(p2):pokep2,peek
    (p2+5)
160 pokep2+5,c:next
161 pokep1,203:pokep1+1,204:pokep1+
    2,205:pokep1+3,204:pokep1+4,206
162 pokep1+5,204:pokep1+6,207:pokep
    1+7,208
163 fora=0to300:next:geta$:ifa$<>""
    then169
164 pokep1,192:pokep1+1,193:pokep1+
    2,194:pokep1+3,193:pokep1+4,195
165 pokep1+5,193:pokep1+6,196:pokep
    1+7,197
166 fora=98to75step-1:forb=0to7:poke
    v1+2*b+1,a:next:c=peek(p2)
167 pokep2,peek(p2+5):pokep2+5,c:next
168 geta$:ifa$=""then159
169 pokev1+21,0:pokev2+21,0
170 print"[CLR-HOME]":fori=1024toi+
    199:pokei+54272,6:pokei,160
171 poke54272+i+800,6:pokei+800,160
    :next
172 print"[HOME][neer][1.blauw][18xrech
    ts][RVS-aan]s.i.t[RVS-uit]"
173 print"[8xneer][6xrechts]u heeft
     net de een na laatste"
174 print"[neer][15xrechts]truc gez
    ien."
175 print"[3xneer][10xrechts]nu vol
    gt de laatste!"
176 fora=0to5000:next
177 .0:print"[CLR-HOME][neer][4xspatie]
    **** commodore 64 basic v2 ****"
178 print"[neer] 64k ram system[2xspati
    e]38911 basic bytes free"
179 print"[2xneer]ready."
180 poke40639,1:fora=1to6:next:poke
    53280,14:poke53281,6
181 print"[5xneer] wilt u deze demo
     nogmaals zien[2xspatie](j/n)":
    poke198,0
182 poke40045,4:.4:geta$:ifa$<>"j"then
    poke53280,14:end
183 f1=1:poke40045,2:poke39712,1:goto
    2
```

*** EINDE LISTING ***

demo sit

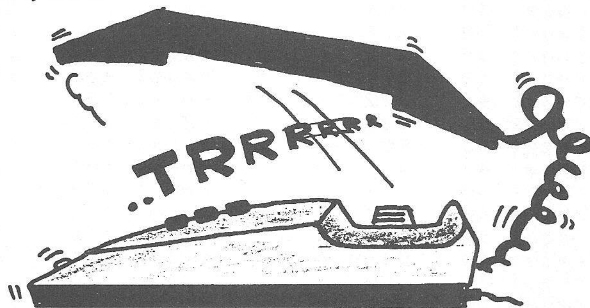
regel 1	74	regel 77	116
regel 2	102	regel 78	26
regel 3	86	regel 79	138
regel 4	200	regel 80	22
regel 5	154	regel 81	207
regel 6	213	regel 82	27
regel 7	40	regel 83	190
regel 8	50	regel 84	248
regel 9	149	regel 85	190
regel 10	175	regel 86	25
regel 11	213	regel 87	78
regel 12	24	regel 88	237
regel 13	178	regel 89	187
regel 14	99	regel 90	200
regel 15	233	regel 91	204
regel 16	57	regel 92	245
regel 17	201	regel 93	110
regel 18	1	regel 94	34
regel 30	150	regel 95	31
regel 31	178	regel 96	98
regel 32	198	regel 97	131
regel 33	204	regel 98	202
regel 34	226	regel 99	8
regel 35	2	regel 100	40
regel 36	51	regel 101	108
regel 37	104	regel 102	229
regel 38	203	regel 103	115
regel 39	218	regel 104	240
regel 40	51	regel 105	11
regel 41	222	regel 106	150
regel 42	220	regel 107	12
regel 43	95	regel 108	163
regel 44	4	regel 109	236
regel 45	141	regel 110	28
regel 46	237	regel 111	57
regel 47	187	regel 112	133
regel 48	20	regel 113	138
regel 49	150	regel 114	173
regel 50	34	regel 115	208
regel 51	117	regel 116	44
regel 52	89	regel 117	39
regel 53	112	regel 118	22
regel 54	10	regel 119	179
regel 55	44	regel 120	147
regel 56	56	regel 121	133
regel 57	49	regel 122	23
regel 58	249	regel 123	19
regel 59	179	regel 124	14
regel 60	30	regel 125	155
regel 61	135	regel 126	37
regel 62	51	regel 127	241
regel 63	139	regel 128	144
regel 64	130	regel 129	20
regel 65	46	regel 130	13
regel 66	46	regel 131	190
regel 67	202	regel 132	141
regel 68	222	regel 133	235
regel 69	21	regel 134	85
regel 70	184	regel 135	206
regel 71	10	regel 136	137
regel 72	235	regel 137	14
regel 73	15	regel 138	193
regel 74	146	regel 139	67
regel 75	99	regel 140	126
regel 76	254	regel 141	128

```
regel 142   156
regel 143   165
regel 144   175
regel 145   157
regel 146   112
regel 147   5
regel 148   240
regel 149   30
regel 150   235
regel 151   193
regel 152   190
regel 153   187
regel 154   183
regel 155   80
regel 156   104
regel 157   149
regel 158   198
regel 159   140
regel 160   35
regel 161   158
regel 162   169
regel 163   240
regel 164   193
regel 165   190
regel 166   30
regel 167   22
regel 168   108
regel 169   73
regel 170   46
regel 171   13
regel 172   149
regel 173   185
regel 174   207
regel 175   25
regel 176   201
regel 177   98
regel 178   121
regel 179   162
regel 180   9
regel 181   131
regel 182   119
regel 183   202

ready.
```

Telefoon

Dat een logo niet alleen door animatie, maar ook door geluid tot leven kan worden gebracht laat Jaap Dijkema uit Westzaan in de onderstaande listing zien. Het teken dat op de letter V lijkt in de regels 21, 22, en 23 is het worteltrekteken. Dit verkrijgt u door in de juiste letterset`SHIFT± aapje (naast de letter P) in te tikken. Grafische tekens intikken blijft een lastig karwei, maar als u nieuwschierig genoeg bent naar dit programma, moet het lukken lijkt me.




```
1 rem telefoon 64
2 rem door jaap dijkema
3 rem westzaan
4 rem
5 printchr$(142);chr$(8):si=54272
6 poke53280,12:poke53281,12:print
  "[CLR-HOME][ROOD]";
7 print"[ROOD][GROEN][BLAUW][RVS-aan]
  [39xspatie][RVS-uit]
8 print"[oranje][BLAUW][RVS-aan][4xsp
  atie]even bellen met commodore-
  info[5xspatie][RVS-uit]
9 print"[oranje][BLAUW][RVS-aan][39xs
  patie][RVS-uit]"
10 print"
11 print"[ROOD][3xspatie]UCCCCI
12 print"[ROOD][2xspatie]U┤[4xspatie]
   ├CCCCCCCI
13 print"[ROOD][2xspatie]BB[4xspatie]
   B[2xspatie]ptt[2xspatie]B
14 print"[ROOD][2xspatie]B├I[2xspatie]
   U┤UCCCCCIB[5xspatie][WIT] O----
   ---------P"
15 print"[ROOD][2xspatie]BBB[2xspatie]
   BB [ZWART]▒▒▒▒▒[ROOD] B[5xspatie]
   [WIT] |post[7xspatie][RVS-aan][GEEL
   ][2xspatie][RVS-uit][WIT] |P ";
16 print"[ROOD][2xspatie]BBB[2xspatie]
   BB [ZWART]▒▒▒▒▒[ROOD] B[5xspatie]
   [WIT] | [11xspatie][RVS-aan][GEEL]
   [2xspatie][RVS-uit][WIT] | |P";
17 print"[ROOD][2xspatie]BBB[2xspatie]
   BB[7xspatie]B[5xspatie][WIT] | [13xs
   patie] | | |";
18 print"[ROOD][2xspatie]BBB[2xspatie]
   BB[2xspatie][ZWART][RVS-aan]123
   [RVS-uit][ROOD][2xspatie]B[5xspatie
   ][WIT] | [5xspatie][ZWART]com-in
   fo[WIT] | | |";
19 print"[ROOD][2xspatie]BBB[2xspatie]
   BB[2xspatie][ZWART][RVS-aan]456
   [RVS-uit][ROOD][2xspatie]B[5xspatie
   ][WIT][shift-SPATIE]| [ZWART]:::
   : pb[2xspatie]112 [WIT] | | |";
20 print"[ROOD][2xspatie]BBB[2xshift-S
   PATIE]BB[2xspatie][ZWART][RVS-aan]
   789[RVS-uit][ROOD][2xspatie]B[5xspa
   tie][WIT] | [ZWART]:::: blaricum
   [WIT] | | |";
21 print"[ROOD][2xspatie]BBB[2xspatie]
   BB[2xspatie][ZWART][RVS-aan]*0#
   [RVS-uit][ROOD][2xspatie]B[5xspatie
   ][WIT] L______________√| |";
22 print"[ROOD][2xspatie]B├K[2xspatie]
   J┤[5xspatie][ZWART] [ROOD] B[7xspat
   ie][WIT]L______________√|";
23 print"[ROOD][2xspatie]BB[4xspatie]
   B [ZWART] Q Q [ROOD] B[8xspatie]
   [WIT]L______________√";
24 print"[ROOD][2xspatie]J┤[4xspatie]
   ├CCCCCCCK
25 print"[ROOD][3xspatie]JCCCCK
26 print
27 print
28 print"[bruin]
29 print"[ZWART]ha,hij gaat al ove
   r"
30 print"[ZWART]------------------
   -"
31 pokesi+24,15:fori=0to4:pokesi,4
   0 :pokesi+1,234
32 pokesi+7,130:pokesi+8,110:pokes
   i+14,20:pokesi+15,117
33 pokesi+6,240:pokesi+13,240:poke
   si+20,240:fort=0to500:nextt
34 forj=0to20:pokesi+4,33:pokesi+1
   1,33:pokesi+18,17
35 pokesi+4,32:pokesi+11,32:pokesi
   +18,16:nextj:nexti:pokesi+24,0
36 print"[CLR-HOME][6xneer]"
37 print"[3xrechts][ZWART][2xspatie]
   ja hallo,met dijkema[17xspatie]
   [RVS-uit]
38 print"[2xrechts][WIT][3xspatie]
   dit was even een test[15xspatie]
   [RVS-uit]
39 print"[2xrechts][ZWART][3xspatie]
   mijn vriendelijke dank.[13xspatie]
   [RVS-uit]"
40 get a$:if a$="" then goto 40
```

*** EINDE LISTING ***

telefoon

regel 1	85	regel 5	95
regel 2	212	regel 6	140
regel 3	252	regel 7	184
regel 4	143	regel 8	195

```
regel 9    33
regel 10   187
regel 11   129
regel 12   40
regel 13   215
regel 14   213
regel 15   88
regel 16   188
regel 17   101
regel 18   24
regel 19   151
regel 20   105
regel 21   79
regel 22   111
regel 23   248
regel 24   31
regel 25   120
regel 26   153
regel 27   153
regel 28   80
regel 29   227
regel 30   196
regel 31   157
regel 32   45
regel 33   135
regel 34   175
regel 35   163
regel 36   214
regel 37   54
regel 38   158
regel 39   47
regel 40   186

ready.
```

```
31 print"[neer][11xspatie]gewaarsc
   huwd!"
32 print"[neer][11xspatie]slechts
   eenmaal de"
33 print"[neer][11xspatie]spatieba
   lk in te drukken."
34 print"[neer][11xspatie]nu is uw
   kans helaas"
35 print"[neer][11xspatie]verkeken
   en zult u"
36 print"[neer][11xspatie]uw verde
   re leven in"
37 print"[neer][11xspatie]armoede
   moeten slijten!!"
38 print"[neer][11xspatie]jammer m
   aar helaas."
39 goto39

*** EINDE LISTING ***
```

Lotto 64

Vond u de reeds eerder gepubliceerde lottoprognose te lang om in te tikken en heeft u de listcassette vergeten te bestellen? Hier zolang een noodoplossing. Dit programma maakt geen prognose, maar gaat uit van een andere tacktiek, geluk dus. Als u het bij de lotto nog nooit won, dan is dit misschien de oplossing voor u. O ja mocht u dankzij dit programma rijk worden, denkt u dan ook even aan ons en aan Dhr van Wijk uit Leeuwarden? Dank u.

```
 1 rem lotto 64
 2 rem door h vd wijk
 3 rem leeuwarden / 058-880985
 4 rem
 7 poke53280,6
 8 print"[CLR-HOME]":print"[14xspatie]
   [RVS-aan]lottospel"
 9 print"[GEEL][4xneer][8xspatie]d
   it is uw kans op rijkdom."
10 print"[2xneer][8xspatie]door sl
   echts eenmaal"
11 print"[2xneer][8xspatie](en bes
   list niet vaker)"
12 print"[2xneer][8xspatie]de spat
   iebalk in te"
13 print"[2xneer][8xspatie]drukken
   ,lacht vrouwe"
14 print"[2xneer][8xspatie]fortuna
    u toe!!"
15 geta$:ifa$=""then15
16 ifa$<>chr$(32)then15
17 print"[CLR-HOME][4xneer]"
18 r=0:forb=1to6
19 t=int(rnd(1)*41)+1:l(b)=t:fora=
   1to6
20 ifl(b)=l(a)thenr=r+1
21 nexta:ifr=>7then17
22 print"[neer][10xspatie]";b;"e g
   etal=";l(b)
23 nextb
24 print"[3xneer][8xspatie]ziezo d
   e poet is binnen!"
25 geta$:ifa$=""then25
26 print"[CLR-HOME]":fora=1to15:poke
   53280,1:poke53281,1
27 print"[10xneer][12xspatie][ROOD]
   a-a-i-i!!! n-e-e-!!!"
28 forb=1to6:next:print"[CLR-HOME]"
   :poke53280,6:poke53281,6:next
29 print"[CLR-HOME]":print"[neer][GEEL
   ][11xspatie]niet doen!!!!"
30 print"[neer][11xspatie]ik heb u
    nog zo"
```

```
lotto 64

regel 1    139
regel 2    218
regel 3    186
regel 4    143
regel 7    251
regel 8    95
regel 9    242
regel 10   60
regel 11   162
regel 12   152
regel 13   147
regel 14   157
regel 15   51
regel 16   222
regel 17   180
regel 18   238
regel 19   13
regel 20   210
regel 21   131
regel 22   243
regel 23   196
regel 24   217
regel 25   52
regel 26   186
regel 27   99
regel 28   13
regel 29   16
regel 30   51
regel 31   148
regel 32   128
regel 33   64
regel 34   180
regel 35   128
regel 36   184
regel 37   14
regel 38   167
regel 39   245

ready.
```

Adresbestand

U merkt aan de steeds langer wordende dagen dat de feestdagen weer naderen. Als u net als ik weer op zoek gaat naar uw sjoemelige adresboekje, voor de adressen van uw wenskaarten, moet u toch eens overwegen of dat ding niet aan pensioen toe is. U heeft tenslotte uw homecomputer en hier ligt nu een van die kleine nuttige toepassingen. Minder dan veertig regels tikwerk moet u toch over hebben voor een mooi programma dat al uw adressen bijhoudt. Als u nu begint met tikken bent u nog voor de 'kaartenstorm' klaar.

```
 1 rem adresbestand 64
 2 rem door tjasso roossien
 3 rem groningen / 050-135664
 4 rem
10 poke53281,11:poke53280,13:dima$
   (100),b$(100),c$(100),d$(100),e
   $(100),f$(100)
20 print"[CLR-HOME][neer][rechts][WIT]"
   ;t;"adressen":o$="[CLR-HOME]opz
   oeken!":i$="[CLR-HOME]invoeren!"
   :bl$="[CLR-HOME]bladeren!"
30 print"[neer][rechts]1..........
   adres invoeren":print"[neer][rechts
   ]2..........adres opzoeken"
```

```
 40 print"[neer][rechts]3...bladere
    n door bestand":print"[neer][rechts
    ]4.....opslaan van bestand"
 50 print"[neer][rechts]5.......lad
    en van bestand":print"[neer][rechts
    ]6.................printen"
 60 print"[neer][rechts]7..........
    ......geheugen":print"[neer][rechts
    ]8...................stoppen"
 70 print"[2xneer]keuze!":poke198,0
    :wait198,1:getq:onqgosub80,120,
    150,180,220,260,290,310:goto20
 80 t=t+1:printi$:input"[2xneer]ach
    ternaam";a$(t):input"[neer]voor
    naam";b$(t):print"[neer]adres";
 90 inputc$(t):input"[neer]postcode"
    ;d$(t):input"[neer]woonplaats";
    e$(t):print"[neer]telnummer";
100 inputf$(t):print"[2xneer]ok zo(
    j/n)":poke198,0:wait198,1:getq$
    :ifq$="j"thenreturn
110 t=t-1:goto80
120 printo$:input"[2xneer]welke ach
    ternaam";z$:forx=1tot:ifa$(x)<>
    z$thennextx:return
130 print"[CLR-HOME][2xneer]";b$,a$
    :print"[neer]"c$:print"[neer]"d
    $,e$:print"[neer]"f$
140 print"[2xneer]druk een toets":poke
    198,0:wait198,1:nextx:return
150 fors=1tot:printbl$:print"[2xneer]
    achternaam ";a$(s):print"[neer]
    voornaam ";b$(s)
160 print"[neer]adres ";c$(s):print
    "[neer]woonplaats ";d$(s),e$(s)
    :print"[neer]tel.nummer";f$(s)
170 print"[2xneer]druk een toets":poke
    198,0:wait198,1:nexts:return
180 print"[CLR-HOME]saven!":print"[2xne
    er]druk op toets als cassette k
    laar is.":poke198,0
190 wait198,1:open1,1,2,"ad file":print#
    1,t:fors=1tot:print#1,a$(s):print#
    1,b$(s)
200 print#1,c$(s):print#1,d$(s):print#
    1,e$(s):print#1,f$(s):nexts:close
    1
210 print"[neer]klaar!!":druk een to
    ets voor menu":poke198,0:wait19
    8,1:return
220 print"[CLR-HOME]laden!":print"[2xne
    er]druk op toets als cassette k
    laar is.":poke198,0
230 wait198,1:open1,1,0,"ad file":input#
    1,t:fors=1tot:input#1,a$(s):input#
    1,b$(s)
240 input#1,c$(s):input#1,d$(s):input#
    1,e$(s):input#1,f$(s):nexts:close
    1
250 print"[neer]klaar!!":druk een to
    ets voor menu":poke198,0:wait19
    8,1:return
260 print"[CLR-HOME]":input"[2xneer]
    welke achternaam";p$:forx=1tot:
    ifa$(x)<>p$thennextx:return
270 open4,4:cmd4:print#4,b$(x),a$(x
    ):print#4:print#4,c$(x)
280 print#4:print#4,d$(x),e$(x):print#
    4:print#4,f$(x):print#4:close4:
    nextx:return
290 print"[CLR-HOME][2xneer]geheuge
    n!":print"[2xneer]aantal vrije
    geheugenplaatsen : ";abs(fre(0)
    )
300 print"[2xneer]druk een toets!":
    poke198,0:wait198,1:return
310 print"[CLR-HOME][2xneer]stoppen
    !":print"heeft u het bestand ge
    saved?":poke198,0:wait198,1
320 getq$:ifq$="j"thenprint"[CLR-HOME]
    tot ziens":end
330 print"[2xneer]dan moet u dat no
    g even doen":fore=1to1500:next:
    return

*** EINDE LISTING ***
```

```
adresbestand

regel 1    105          regel 170   13
regel 2    9            regel 180   235
regel 3    96           regel 190   72
regel 4    143          regel 200   206
regel 10    224         regel 210   14
regel 20    82          regel 220   210
regel 30    250         regel 230   10
regel 40    141         regel 240   126
regel 50    117         regel 250   14
regel 60    30          regel 260   114
regel 70    241         regel 270   1
regel 80    66          regel 280   136
regel 90    207         regel 290   186
regel 100    193        regel 300   31
regel 110    97         regel 310   110
regel 120    34         regel 320   225
regel 130    10         regel 330   62
regel 140    18
regel 150    85
regel 160    171        ready.
```

CES 1986

Nederlandse deelname aan Amerikaanse beurs voor consumenten electronica

De Nederlands-Amerikaaanse kamer van koophandel, in samenwerking met de EVD bereidt een gezamenlijke deelname van Nederlandse bedrijven aan de CES (9-12 januari Las Vegas USA) voor. Dat is een ideale springplank om bv. computersoftware of randapparatuur onder de aandacht van de Amerikaanse pers en inkopers te brengen.

Er is de mogelijkheid van subsidie en belangstellenden kunnen contact opnemen met de heer van Dijk. ☎ 070-478234.

MISSERS

Onze grootste misser was, dat op de vorige omslag bij een deel van de oplage Nr 3, 1985 stond, terwijl het heel duidelijk nr 9 had moeten zijn. We hebben een deel laten herdrukken, maar we wilden toch de abonnee's hun blad niet een week later dan normaal toezenden, dus vandaar.

In Commodore-Info no 9 is in het C-16 programma Crazy Sprite op pag 61/62 is als gevolg van regelhernummering een foutje ontstaan. Regel 190 moet worden:

```
190 PRINT F$":(neer)":print (l/
    32-1512)*10 + 10 "data"
    D$"," ch:end
```

C 16 Database

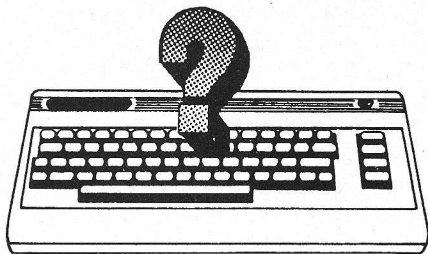
In het vorig nummer hebben we bij het programma C-16 database niet vermeld, dat het programma van T.Kusters uit Amstelveen was. Bovendien bleken de tokenized basic commando's LOOP en CHAR niet goed door te komen. In regel 500 moet dus CHAR staan ipv. SAVE, dat ook in 550 en in 570 hoort CLOSE eigenlijk LOOP te zijn.

In **Videotrack** 64 uit nr 7, 1985 zit een fout, die we al eerder signaleerden, maar waarover veel vragen komen. In regel 20050 moet tussen de " " in de q1$ namelijk een reverse b staan, maar onze printer kende dat teken helaas niet.

In het artikel van Peter Kremer met **Tips&Trucs** uit nr. 8 stonden in de listing een aantal S'en, die natuurlijk $'s (dollartekens) moesten zijn.

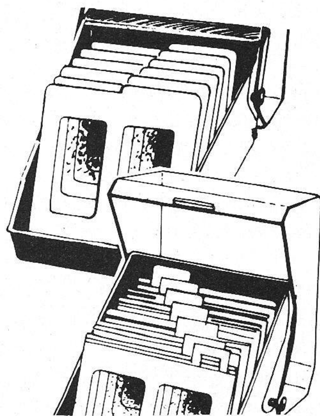
In **woestijnrace** op pag 57 uit nr. 9 staat in regel 10 een teken, dat shift en pijl omhoog betekent. Verder staan in regel 370 twee verschillende soorten streepjes. De ene is het min teken en de andere bereikt u d.m.v. shift-ster. Niet uitzonderlijk, maar wel voor sommige lezers even puzzelen.

Vragen van Gebruikers.



Deze keer vooraf een vriendelijk, maar toch wel dringend verzoek aan de vragenstellers, die graag antwoord willen hebben op hun computerproblemen. Het komt helaas nog al eens voor, dat er een brief naar ons toe wordt gestuurd, zonder dat direkt duidelijk is, dat het epistel bestemd is voor deze rubriek. Dan moet bij het sorteren van de dagelijkse post eerst de hele brief worden doorgelezen, voordat we te weten komen dat het hier een vraag voor de vragenrubriek betreft. En dat kost meer tijd dan nodig is. Daarom aan iedere inzender nogmaals het verzoek, vermeldt duidelijk op de envelop, liefst linksboven, dat het gaat om de vragenrubriek. Dan loop je geen kans, dat de brief naar een verkeerde afdeling verdwijnt, en misschien nooit meer boven water komt.

Verder worden hier alleen die problemen behandeld, die voor meer dan één lezer van belang kunnen zijn. Specifieke vragen over de aansluiting van b.v. een minder gangbare printer of het schrijven van complete programma's voor een bepaalde toepassing kunnen we helaas niet honoreren, al zou ik dat misschien nog zo graag willen, het kost simpelweg teveel tijd.
Daarnaast begrijpt iedereen, hoop ik, dat het ook niet ons werk is om de 'bugs' uit de ons toegestuurde listings (soms langer dan 300 regels) te halen, zodat een zelf geschreven programma foutloos kan runnen. We publiceren in de listingrubriek genoeg utilities om zelf te kunnen kombineren en daarmee goed werkende programma's te kunnen schrijven.



De 1570-diskdrive

Van diverse kanten zijn er vragen binnengekomen, over de nieuwe Commodore-diskdrive, de 1570. Er blijken nogal wat verschillende ideëen over te bestaan, met name over het nut van de nieuwe diskdrive bij gebruik met de C-64. Zo schrijft Jan F. van Struiken: Als ik nu een 1570-drive aanschaf, in plaats van de 1541 waarvoor ik zo lang heb gespaard, kan ik dan ook sneller werken met de 64, net als bij de C-128 ?

De nieuwe diskdrives, die door Commodore in Europa op de markt worden gebracht, of binnenkort (?) nog komen, zijn in principe bedoeld voor de C-128. De beide types, de 1570 en 1571 zijn wat hardware en operating systeem betreft, nagenoeg identiek. Bij de 1571 zitten er alleen twee lees/schrijfkoppen in, zodat deze dubbelzijdig kan werken, en dus een dubbele opslagcapaciteit heeft. In snelheid is er geen verschil tussen deze modellen. Ze zijn beide in staat om diverse CP/M-formaten te lezen, en daarnaast het Commodore 4040 (35 tracks) formaat, zoals ook de 1541 doet.
De 1570 wordt inmiddels verkocht, als voorloper van de 1571, die pas in mei 1986 in Europa wordt verwacht.
Ik heb van beide drives een paar apparaten kunnen testen, en deze werken met de C-128 inderdaad supersnel, als je dit tenminste vergelijkt met de 1541.
Voor de C-64 kun je de 1570/71 wel gebruiken, maar, omdat het operating-systeem t.o.v. de 1541 nogal verschilt, zal niet alle beschermde C-64 software goed functioneren met de 1570/71. De snelheid van de 1570 wordt dan, net als bij de C-128 in de 64- stand helaas ongeveer de zelfde als die van de 1541. Als je dus sneller wilt werken met de C-64 en de diskdrive, helpt het niet om je 1541 te verwisselen voor een 1570. Jammer, maar de in de C-64 aanwezige I/O is nu eenmaal qua seriële overdracht niet ontwikkeld om snel te kunnen communiceren, ook niet met de 1570/71.

Tape-compilers

A. van Oorschot uit Doetichem vraagt of er voor de C-64 Basic compilers te krijgen zijn, die met de datasette werken.

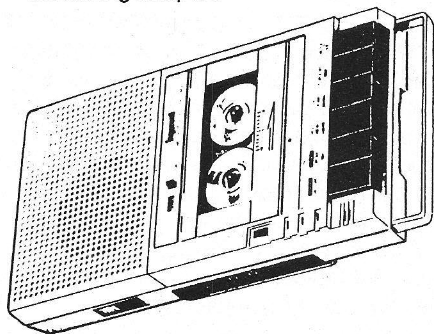
Zover mij bekend zijn er geen cassettecompilers op de markt. Compilers werken meestal in een paar gangen een bepaald Basic-programma door, om er de uiteindelijke compacte, snelwerkende versie van te fabriceren. Daartoe zit het compiler- programma in het computergeheugen en het programma wordt een paar keer in z'n geheel van de disk gelezen, terwijl ook de nodige hulpbestanden op de disk worden weggeschreven. Je begrijpt, dat dit met tape niet uitvoerbaar is.
Het feit dat de compilers uitsluitend met een diskdrive werken, betekent echter niet, dat je er geen programma's op kunt compileren die voor tape geschikt zijn. In jouw geval zul je dus een vriend, of een naburige C-64 club, moeten zoeken, die beschikt over een diskdrive èn een compiler. Daarmee kun je dan je eigen programma's bewerken en vervolgens thuis weer met je recorder gaan gebruiken.

Listings

Jaques de Belder uit Antwerpen schrijft in een lange brief, dat hij nogal wat problemen heeft met het lezen van de listings uit ons blad. Zo kan hij b.v. soms niet goed bepalen, welk grafisch teken er precies wordt bedoeld, terwijl er aan het einde van de regels soms een paar plaatsen leeg zijn, waarvan niet duidelijk is, volgens hem, of ze nu wel of niet als spaties moeten worden geinterpreteerd.

Iedereen die zelf weleens heeft geprobeerd, om een leesbare listing van z'n programma's op de printer te krijgen, weet hoe moeilijk dat kan zijn. Dit komt m.n. door de grote hoeveelheid grafische- en controle-tekens, die in de Basic van Commodore kunnen worden toegepast. Ik ben van mening, dat al deze tekens voor de duidelijkheid **moeten** worden omschreven in (CHR$)-waarden. Maar het merendeel van de programma's die we voor publicatie krijgen toegestuurd, zitten vol met grafische tekens en soms zelfs heel vreemde dataregels, die voor de overtypers de nodige ergenis opleveren. Het is duidelijk, dat het intypen van dit soort regels voor de auteurs gemakkelijker is, maar erg korrekt is het niet.
De oplossing voor dit probleem kun je alleen vinden door het zelf uittesten van de diverse grafische tekens, tot je de goede gevonden hebt. In de listings staan ze precies zo als op de toetsen van je 64. In dit verband zou ik de Basic-schrijvers wil-

len vragen hun programma's zo te maken, dat ze alle grafische tekens weergeven als CHR$-strings met de betreffende ASCII-waarde. En DATA-regels ook graag met leesbare (Hex) getallen. Daar zijn we allemaal mee geholpen.



Als er in een Basic-regel, tussen aanhalingstekens méér dan 1 spatie of ander karakter voorkomt, dan vertaalt Visi-list dit zonder meer in (5x spatie) e.d. De lege plaatsen aan het einde van de gedrukte listings moeten dus niet worden overgetypt, ze staan er alleen om de listing wat overzichtelijker te maken.

Geheugenruimte

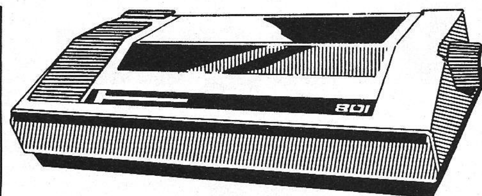
Y.Kinnaer vraagt welke geheugenplaatsen in het RAM/ROM gebied van de C-64 gebruikt kunnen worden voor het wegzetten van machinetaal-programma's, en welke het meest geschikt zijn.

In de C-64 zit een erg goed doordachte struktuur, waar het de beschikbare geheugenruimte betreft. Zoals iedereen weet, beschikt het apparaat over maar liefst 64 KBytes RAM-geheugen. Daarnaast zit er ook nog de nodige (vaste) ROM-I/O-KERNAL ingebouwd. Omdat we te maken hebben met een 8-bits machine, kan door de processor met de beschikbare 16-bits (2x8) 'slechts' 64K aan geheugenadressen direkt worden aangesproken. Deze adressen liggen tussen $0000 en $FFFF. Hierin moet natuurlijk de ROM een plaats hebben. Bij de C-64 bevindt de Basic-interpreter zich van $A000 (40960) tot $BFFF (49151). De VIC,SID en I/O zit van $D000 tot $DFFF, in decimaal 53248 - 57343. Voor de KERNAL zijn de bovenste 8K gereserveerd, van $E000 (57344) tot $FFFF (65535).
In principe kan de rest van de C-64 zonder meer gebruikt worden voor machinetaal-programma's. Maar als er gelijktijdig met Basic moet worden gewerkt, gaat er het nodige vanaf. De geheugenruimte voor scherm en spritepointers zit op $0400(1024) tot (globaal) $07FF (2047). Basic-programma's beginnen meestal op $0800 (2048) en kunnen met het geheugen nodig voor de variabelen doorgaan tot $9FFF (40959), in totaal precies 38911 Bytes, het aantal dat je bij het opstarten van de C-64 als Bytes Free ziet staan.
Om ergens in het 64-geheugen machinetaal-programma's of -routines weg te zetten, kun je een deel van de Basic-RAM afhalen, door vooraf de pointer, die de top van het vrije geheugen aangeeft met een lager getal te poken. Deze pointer zit op adres ($37) 55/56. Geef je dus b.v. POKE 55,0:POKE 56,128, dan betekent dit, dat je Basic geheugentop nu 128 x 256 = 32768 ofwel $8000 is geworden. Minder Bytes Free in Basic, maar je hebt nu een ruimte voor MT- programma's van totaal 8192 Bytes, tussen $8000 en $9FFF zonder andere C-64 mogelijkheden in de weg te staan.
Voor kleine MT-routines is het mogelijk om de cassette-buffer te gebruiken, als je tenminste niet met de datasette werkt. Deze buffer is 191 Bytes groot, begint op $033C (826) en loopt tot aan het Videogebied ($0400). Daarnaast natuurlijk het door bijna alle MT-programma's gebruikte gebied van $C000 (49152) tot $CFFF (53247). Omdat ook de meeste toolkits en MT-assemblers etc. van deze ruimte gebruik maken is deze denk ik minder geschikt voor programma's die je zelf wilt gaan schrijven. Ook de schaduwRAM **onder** de Basic-ROM kan natuurlijk worden gebruikt, maar daarbij moet je er rekening mee houden, dat je niet gelijktijdig met Basic en Machinetaal kunt werken, omdat je in dat geval het geheugen moet schakelen (Bank-switching) tussen Basic en Machinetaal.

Printer test

Ardeel Walter uit Eekloo, (Belgie?) vraagt of er een goede manier is, om in Basic programma's te kunnen testen of de printer aanwezig is en aanstaat, zonder op het scherm de vervelende 'Device not present'-fout te krijgen, waarna je weer overnieuw moet beginnen met je programma.

Ardeel, je zult ongetwijfeld niet de enige zijn, die met dit probleem te kampen heeft. Het is inderdaad minder fraai, als je vergeten bent de printer aan te sluiten, en je programma gaat verkeerd doordat via de foutmelding de zaak in de READY-mode terecht komt.
Een echte Basic-test heb ik niet voor elkaar kunnen krijgen, want de interpreter test bij iedere print-opdracht de I/O status (variabele ST) zonder dat je daar als programmeur tussen kunt komen. Maar wel heb ik een kleine machinetaalroutine gemaakt, waarmee je uit de voeten kunt. Zet



hem voor in je programma en iedere keer, nadat je een OPEN opdracht geeft kun je m.b.v. **SYS(830)** het ST-getal PEEKen op 144. Is dit getal < > 128 dan kun je er van op aan dat de printer goed staat. Bij 128 betekent het, dat het 7e bit van ST gezet is, dus een fout in de printeraansturing. Basic geeft op grond hiervan een 'DEVICE NOT PRESENT' error, maar deze wordt omzeild door dit stukje machinetaal:

```
10 REM PRINTER-TEST
20 DATA 169,4,32,180,255
30 DATA 32,183,255,96
40 FOR I = 830 TO 838
50 READ A : POKE I,A
60 NEXT
70 REM OPEN EN TEST PRINTER
80 OPEN 4,4
90 SYS (830): REM DE TEST
100 IF PEEK(144) < > 128 THEN 150
110 REM PRINTER FOUT
120 PRINT "CURSOR OP± PRINTER
    AANZETTEN"
130 CLOSE 4: GOTO 80
140 REM PRINTER OK
150 PRINT#4," DE PRINTER WERKT"
etc....
```

In principe werkt deze routine voor **alle** printers op de C- 64. De getallen in adres 144 ($90) kunnen afwijken, maar daar is door wat proberen wel uit te komen. De MT-routine is naar ieder adres in het geheugen te verplaatsen, het aanroepadres via SYS moet dan natuurlijk ook worden aangepast.

Shift-return

Van diverse kanten heb ik problemen opgevangen over de betekenis van bepaalde SHIFT-karakters. Zo heb je bij alle Commodore-computers naast de normale spatie en RETURN ook een SHIFT-versie. H.Verhorst vroeg expliciet naar de verschillen.

De ASCII karakterset van de Commodore computers bestaat uit precies 256 tekens per mode. Deze tekens komen voor een deel dubbel voor, een ander deel kan zowel met als zonder SHIFT worden gebruikt. Het verschil bij het gebruik van de SHIFT-toets is bij de alfanumerieke toetsen denk ik wel duidelijk. Met de RETURN-toets en de spatiebalk ligt dat anders. In feite zijn het gewone tekens, waar je weinig mee kunt doen. De shift-spatie, met een ASCII-waarde van 160 reageert net zo als

een normale spatie (ASCII 32), je ziet hem niet als je hem print. De ASCII waarde die je uit de string terug kunt lezen met ASC("SHIFT SPATIE±") blijft echter altijd 160. Dit kan je in bepaalde gevallen helpen.
Er zijn nogal wat (niet Commodore) printers die de normale spatie soms negeren als begin-Bytes in een string. Maak je daar nu 'shift-spaties' van, dan worden ze ook door de printer als volwaardige karakters gezien en als spaties op papier afgedrukt. De shift-return is feitelijk net zoiets. Druk je hem in op het toetsenbord, dan reageert het scherm als was er een return getypt. Maar de programma-regel wordt op dat moment niet in het geheugen gezet. Bij het schrijven van Basic dus handig als je met de cursor uit een foute regel wilt springen. ASCII waarden RETURN = 13, Shift-return = 141.

Verplaatsen Machinetaal

Het regent vragen uit Belgie deze maand. Uit Roeselaere komt van Lambert Rosa een heel epistel met opmerkingen en vragen over het verplaatsen van Machinecode in het computergeheugen. Wat hij er precies mee wil is niet erg duidelijk, wel komt eruit, dat de code die hij heeft verplaatst niet altijd evengoed meer werkt.

Machinecode wordt in een bepaald gedeelte van het geheugen gezet, en werkt dan met bepaalde adressen, zowel in als buiten de code zelf. Dit zijn dan b.v. adressen van de nul-pagina (ZERO- PAGE) of KERNAL routines. Het verplaatsen van zo'n routine kun je vergelijken met het hernummeren van een Basic-programma. De regelnummers veranderen en dus moeten ook de GOTO en GOSUB opdrachten worden aangepast. De eventueel aanwezige SYS-opdrachten moeten echter gelijk blijven, dat zijn de adressen **buiten** de routine.
Er zijn de nodige M/C-relocaters verkrijgbaar. Ook in C-info jaargang 1 nummer 3 stond een korte, goede Basic-listing van zo'n Code-verplaatser. Het probleem blijft bij al die hulpmiddelen echter bestaan, dat niet alle Machinetaal moet en kan worden verplaatst. Er zitten in goed geschreven code nogal eens z.g. woord-tabellen of tekst-strings, die voor zo'n programma lijken op normale code, dus kan het niet anders of daar worden fouten gemaakt. Je kunt beter gebruik maken van een assembler en je software met labels schrijven, zodat je daarna het programma in M/C op iedere gewenste plek in het geheugen kunt neerzetten. Het lijkt me duidelijk dat het bijna onbegonnen werk is om (commerciele) machinecode van anderen te gaan relokeren.

Oude C-64 schermpokes

Dhr. J.Vernig uit Maarsbergen heeft een probleem met de manier waarop wij sommige listings plaatsen. We houden geen rekening met de lezers die beschikken over het oudste type C-64, waar het poken naar 't scherm wat ingewikkelder is.

Toen de eerste C-64 machines werden gebouwd zaten daar nogal wat soft- en hard-bugs in. De meeste van deze serie zijn in Europa dan ook niet verkocht. Een deel van deze machines, waaronder die van dhr. Vernig hebben toch de consument bereikt. Het fenomeen waarover hij klaagt zit in de Video-Interface Chip. Bij de oude C-64 moet bij het poken naar de schermlokaties ook dezelfde lokatie van de kleurenRAM worden gepoked in de gewenste kleur. Anders krijgt het karakter de achtergrondkleur, en dan zie je hem niet. Het gebied van de kleurenRAM zit op de adressen 55296 tot 56295.
Heb je zo'n C-64, en lijkt het alsof een programma niet werkt, probeer dan bij alle pokes naar de adressen 1024 tot 2023 de toevoeging te maken van POKE (55296+scherm-1024) , kleur.
Wil je een test welk type C-64 je hebt, probeer dan het miniatuurtje uit C-info jrg 2 nr.8, dat de tien met een griffel kreeg, onverdiend volgens de heer Vernig.

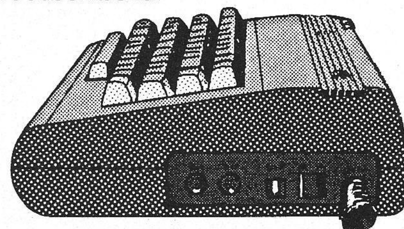
Barcode



C.Molenaar uit Eelde vraagt of het niet mogelijk is om de Basic- listings in het vervolg in barcode te publiceren. Dat zou veel handiger zijn met het overzetten naar de eigen machine.

Het lijkt op zichzelf erg handig, om met de lichtpen even over een bladzij van Commodore Info te strijken en vervolgens RUN te typen op je C-64 en ziedaar, wat net nog als een onbegrijpelijk streepjes-patroon op papier stond, komt nu als een prachtig programma tot leven in de computer van meneer Molenaar.
Een prima idee, ware het niet dat er nogal wat minder aardige kanten aan zitten. Dit idee zou niet misstaan hebben in onze ideeën prijsvraag over de toekomst van de computer. Vooralsnog is het echter zo, dat een barcode niet zo erg veel informatie kan bevatten. Daarom zou een Basic-listing van zo'n 20 regels meer dan twee pagina's van C-Info kosten. Op de twaalf listing pagina's kunnen we in dat geval slechts een programma afdrukken van minder dan 9600 Bytes, ofwel 10% van wat er nu in die rubriek te vinden is. U weet toch, dat alle listings, die in ons blad worden geplaatst ook op tape of disk te krijgen zijn bij Infolist. Dat gaat nog sneller en bespaart U ook nog het lezen en vertalen van de streepjescode.

Toetsenbord

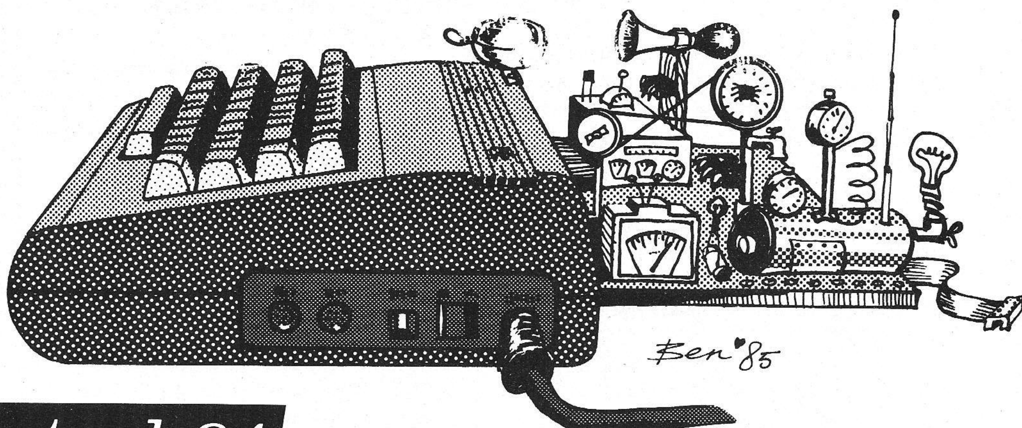


J.de Haan uit Zwolle heeft problemen met z'n toetsenbord van de VIC-20. Na jarenlange trouwe dienst zijn er bepaalde veel gebruikte toetsen die het soms af laten weten. Als je erg hard drukt, zo schrijft meneer de Haan, dan heb je kans dat de letter nog op het scherm komt, maar meestal lukt dat niet. Wat is hier aan te doen?

Het feit, dat bij de oudere computers niet alle toetsen meer even goed functioneren is te wijten aan een vervuiling van de toetskontakten die onderin het toetsenbord zitten. Om hier iets aan te doen, moet de bovenkant van de computer worden losgeschroefd en daarna het toetsenbord van de bovenkant worden los gehaald. Denk erom, dat hier de 4 goede schroeven worden losgedraaid, anders vliegen de toetsen en de veertjes overal heen en wordt het een hele puzzel alles weer op z'n plek te krijgen. Daarna kunnen de toetscontactjes met contact-spray worden bespoten. Als de zaak weer in elkaar is gezet, werken alle toetsen weer zoals ze moeten. Is er iets in het toetsenbord (bijna) versleten, zoals dat bij de RETURN-toets nog weleens gebeurt, dan kan het soms helpen om het mechaniek te verwisselen met dat van een minder vaak gebruikte toets. Daarvoor moet echter het hele toetsenbord uit elkaar. J.B.

We gaan deze keer verder met de behandeling van de VIC-chip. Hierbij zullen wij ons bezig gaan houden met het gebruik van meerdere kleuren tegelijk, en natuurlijk ook met de populaire sprites. Daarna wordt de analyse van het PLOT-programma vervolgd. We zullen dan zien hoe je de Basic-interpreter kunt kombineren met routines voor zelfgemaakte opdrachten.

Programmeren in machinetaal door Sjoerd Bakker Aflevering 8



Serie Machinetaal 64

EBCM en Sprites

In de vorige aflevering hebben we gewerkt met de meest eenvoudige manier om tekst en grafiek op het scherm zichtbaar te maken. Bij tekst hadden we dan voor het hele beeldscherm dezelfde achtergrondkleur en we konden elk karakter (vakje van 8 bij 8 puntjes) een van de 16 verschillende kleuren geven. In de grafische mode konden we per vakje kiezen uit 16 verschillende achter- en ook uit 16 verschillende voorgrondkleuren.

Voor de meeste zakelijke toepassingen zijn bovenstaande mogelijkheden meer dan toereikend, maar de grafische fijnproever heeft er niet genoeg aan. Om ook hem tevreden te stellen is de Commodore 64 voorzien van een paar extra **Color-Modes.**

Tegenover het voordeel om meerdere kleuren te kunnen gebruiken staat echter ook altijd een nadeel, zoals we zullen zien. Daarin staat de 64 niet alleen, want zelfs bij een supercomputer als de Atari 520ST gaat het kleurgebruik ten koste van de schermresolutie (daarover straks meer).

Extended Background Color Mode (EBCM)

EBCM wordt alleen in de tekst-mode gebruikt, en het dient om per karakter 4 verschillende achtergrondkleuren mogelijk te maken.

EBCM zet je aan door bit 6 van register $D011 "1" te maken.

Als je de karakter-tabellen in de Reference Guide bekijkt, valt het op, dat Commodore's lettercode afwijkt van de ASCII- standaard. De **ASCII-waardes lopen van 64-90: de Commodore-waardes van 0-26.** Dit heeft tot gevolg dat alle belangrijke codes in set 1 kleiner zijn dan 64, zodat ze met 6 bits weergegeven kunnen worden. Je houdt dan per schermgeheugenplaats twee vrije bits (bits 7 en 6) over waar je iets anders mee kunt doen.

In EBCM worden deze bits gebruikt om aan te geven welke achtergrondkleur het betreffende karakter op het scherm krijgt. Die waardes ($00-$11) corresponderen met een van de vier kleurregisters die lopen van $D021-$D024. De voorgrondkleur wordt, zoals gebruikelijk, uit de nybbles van het geheugen gehaald.

In EBCM heb je dus niet meer de beschikking over de (grafische) karakters met een code die hoger is dan 63. Een nadeel, dat in de meeste gevallen nauwelijks als zodanig zal worden ervaren.

Multi Color Mode (MCM).

MCM kun je zowel in de tekst-mode als in de grafische mode gebruiken.

MCM zet je aan door bit 4 van register $D016 "1" te maken.

Hiermee wordt het principe van een achter- (0) en een voorgrondkleur verlaten en kunnen we voor elk vakje van 8 bij 8, per twee horizontale puntjes, kiezen uit 4 verschillende kleuren. De twee puntjes krijgen telkens allebei de kleur die ze samen aangeven, zodat de horizontale resolutie met de helft afneemt. Met andere woorden: het kleinst mogelijke gebied dat je op het scherm kunt adresseren is nu niet 1, maar 2 (horizontale) puntjes groot.

Achtergrondkleur ECBM

EXTENDED BACKGROUND COLOR MODE	
KARAKTER CODE BITS 7/6	ACHERGRONDKLEUR UIT REGISTER
0 0	$D021
0 1	$D022
1 0	$D023
1 1	$D024

Zoals we in de vorige aflevering hebben gezien bestaat een karakter uit 8 bytes, waarvan de bits samen een "0-1"-patroon vormen. Normaal gesproken correspondeert een 0 dan met een bepaalde kleur, en een 1 met een andere. In MCM gaat het anders.
Een byte wordt opgesplitst in vier groepjes van twee bits, waarbij elke groepje vier verschillende waardes kan vormen. Deze waardes corresponderen dan (net als in EBCM) met een bepaalde kleur. In de tekst-mode worden "$00-$10"-kleuren uit de eerste drie kleurregisters ($D021-$D023) gehaald, en de "$11"-kleur wordt bepaald door de waarde van de 3 onderste bits (bits 2-0) van de color nybbles.

```
MCM BITPATRONEN
HOOFDLETTER A

00 01 10 00
00 11 11 00
01 10 01 10
01 11 11 10
01 10 01 10
01 10 01 10
01 10 01 10
00 00 00 00
```

Bitpatroon letter A

Het komt erop neer dat de eerste drie waardes voor het hele scherm dezelfde kleuren geven, en dat kleur vier voor elk vakje verschillend kan zijn.

MULTI COLOR MODE KARAKTERS	
BITPAAR	KLEURREGISTER
0 0	$D021
0 1	$D022
1 0	$D023
1 1	COLOR RAM

Mult Color tekens

De tekens van de Character-ROM zijn niet op MCM gemaakt, dus zul je voor een goed effekt je eigen MCM- karakters (of grafische tekens) moeten ontwerpen.

Grafische MCM

Ook in de grafische mode kun je met meerdere kleuren tegelijk werken. Het werkt hetzelfde als in de tekst-mode, alleen dienen nu niet het schermgeheugen en de Character-ROM als beeldschermbron, maar het grafische geheugen.
Een grafische byte wordt weer opgesplitst in vier bitparen, waarvan de waardes corresponderen met een bepaalde kleur. De $00- kleur is gelijk aan de kleurwaarde in register $D021. De $01- kleur wordt gehaald uit het hoge nybble van de betreffende "schermgeheugenplaats" (het schermgeheugen doet in de grafische mode dienst als kleurgeheugen voor een vakje van 8 bij 8 puntjes).
De $10-kleur correspondeert met het lage nybble van de schermgeheugenplaats, en de 11-kleur wordt uit het betreffende color-nybble gehaald.
De horizontale resolutie wordt weer gehalveerd, want de twee puntjes krijgen beide de kleur die ze samen aangeven. Er zijn dus per vakje van 8 bij 8 32 verschillende adresseringspunten, waarover je maximaal vier verschillende kleuren kunt verdelen. Een ervan is voor het hele scherm gelijk: voor de andere drie kan een keuze gemaakt worden uit de 16 standaardkleuren van de C-64.
Tot zover VIC's tekst, grafiek en kleuren. Het maken van een leuk plaatje is erg arbeidsintensief, maar je hebt dan ook veel voldoening van het resultaat. Er zijn speciale hulpprogramma's voor op de markt, die je in ieder geval het technische werk uit handen nemen, zodat je je alleen maar hoeft te concentreren op het "kreatieve

MULTI COLOR MODE HIRES-SCHERM	
BITPAAR	KLEURREGISTER
0 0	$D021
0 1	HI-NYBBLE 'SCHERMGEHEUGEN'
1 0	LO-NYBBLE 'SCHERMGEHEUGEN'
1 1	COLOR RAM

Hi-Res Multicolor

proces". Nu we toch bezig zijn: een erg goed boek op VIC-gebied vind ik nog altijd "Het Grafiekboek voor de Commodore 64" van Plenge, maar dat is wellicht een kwestie van smaak.

Sprites.

Iedereen kent ze wel, die enge kereltjes, onherkenbare vliegtuigjes en uitgehongerde atleetjes. We hebben het over Sprites, de nieuwe helden van de beeldbuis, die snel en ook geanimeerd over het beeld schuiven. Het lijkt of ze door een magische formule voor eeuwig aan je joystick zitten gekleefd, maar dat is natuurlijk slechts schijn. We zullen hier uit de doeken doen wat er zich werkelijk afspeelt in de wereld van het avontuur op de rechthoek van je beeldbuis. Het principe van verplaatsing op het beeldscherm is iets dat de meeste computers kennen. Denk maar eens aan het **LIST**-effekt, waarbij elke regel een rij naar boven opschuift als het scherm vol is. En in de grafische mode zou het bijvoorbeeld mogelijk zijn om een verzameling 1-bits, die samen een figuur voorstellen, softwarematig één voor één te verplaatsen zodat je een bepaalde beweging ziet. Erg handig is dat niet, want je moet hier veel ingewikkelde berekeningen voor uitvoeren, en ook moet je telkens de oorspronkelijke achtergrond bewaren om die weer te voorschijn te halen als de figuur "voorbij is getrokken".

Vandaar waarschijnlijk dat een verlichte geest op zekere dag de Sprite uitvond. In de Commodore 64 komt het erop neer, dat je zelf 8 gebiedjes van 24 bij 21 puntjes samen kunt stellen, waarvan je elke puntje aan (1) dan wel uit (0) kunt zetten. **De 1-en vormen dan samen een bepaald figuurtje die we een Sprite noemen.** Zo'n Sprite kun je over het beeldscherm bewegen door coördinaatwaardes in de twee (eigenlijk drie) daarvoor bestemde registers te zetten. In plaats van (24*21 =) 504 puntjes zelf één voor één te verplaatsen, neemt VIC deze taak nu voor zijn rekening. Het enige wat je nodig hebt is een x- en y coördinaat. VIC onthoudt automatisch de achtergrond, en zet die weer op de oorspronkelijke plaats terug, als de Sprite naar een andere plaats op het scherm is gegaan.

Lokatie ontwerp

Je wilt dus een sprite ontwerpen. Dan moet je eerst weten waar je het ontwerp neerzet. Een sprite-ontwerp legt beslag op

(504:8=) 63 bytes RAM-geheugen. 63 is natuurlijk een onmogelijk getal, dus maken we er voor het gemak 64 van. Omdat VIC 16K tegelijk kan zien, zijn er (16K:64=) 256 mogelijke plaatsen waar je een ontwerp neer kan zetten. Dat komt mooi uit, want dan kunnen we met de inhoud van één byte bepalen waar in de 16K het ontwerp komt te staan. Die byte noemen we de **Sprite-pointer**. De formule van de lokatie van een ontwerp wordt dan:

Lokatie = Bankbasisadres + (Spritepointer ⭑ 64).

De 8 sprite-pointers beslaan de hoogste 8 bytes van het schermgeheugen. Het schermgeheugen is namelijk steeds 1K=1024 bytes groot, maar op het beeldscherm kunnen maximaal maar 40 ⭑ 25=1000 tekens staan.

Je houdt dus bij elk scherm 24 bytes over. Bytes 1016-1023 ($03F8-$03FF) + het schermbasisadres zijn dan bestemd voor de sprite-pointers van sprites 0-7.

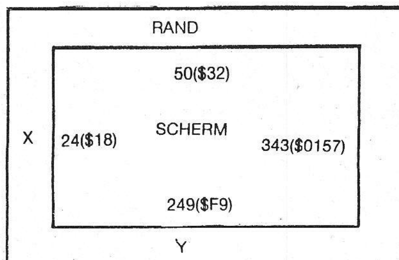
Een voorbeeld. VIC kijkt naar Bank 0, en het scherm begint op 1024, zoals gebruikelijk. Om nu sprite 0 te ontwerpen op de plaats van de cassette-invoerbuffer (828-1019) moet de inhoud van (1024+1016=) 2040 bijvoorbeeld 13 zijn. Het ontwerp begint dan vanaf lokatie $0000+13 ⭑ 64=832=$0340. Dit is dan het spriteontwerp-basisadres.

De bits van lokatie 0 (+spriteontwerp-basisadres) corresponderen met een horizontale rij van 8 puntjes uiterst linksboven in de sprite. Lokatie 1 komt er rechts naast, en lokatie 2 komt weer naast lokatie 1. Lokatie 3 komt onder lokatie 0 te liggen, en zo gaan we door tot we 21 rijen van 3 bytes hebben gehad. Een 1-bit correspondeert met de sprite-kleur en een 0 maakt het betreffende puntje transparant (achtergrond is zichtbaar).

We weten nu waar de sprite staat. Om hem op het scherm zichtbaar te maken moeten we hem aanzetten. De bits van register $DO15 zijn bedoeld om de 8 sprites aan (1) dan wel uit (0) te zetten. Bits 0-7 hebben betrekking op sprites 0-7. Met de volgende routine kan je bijvoorbeeld sprite 0 aanzetten, zonder de andere sprite toestanden te veranderen:

LDA $DO15, ORA #$01, STA $D015.

Je kunt elke sprite ook een eigen **kleur** geven. De kleurregisters die lopen van $D027-$D02E zijn bestemd voor sprites 0-7. In MCM werken we ook bij sprites weer met het bitpaar-principe. MCM zet je aan door per sprite het betreffende bit van register $D01C "1" te maken. Bits 0-7 horen weer bij sprites 0-7.



De **00-kombinatie** laat het betreffende paar transparant; wat er op het scherm staat is zichtbaar.

"01" wordt gehaald uit register $DO25.

"10" neemt de kleur aan van het normale kleurregister.

"11" krijgt de kleur van register $D026.

Sprites kun je twee keer zo lang en/of twee keer zo breed maken. Hiervoor zijn twee **vergrotingsregisters** beschikbaar, waarvan de bits weer betrekking hebben op verschillende sprites. Register $D01D dient voor horizontale, en register $D017 voor vertikale vergroting. Een 1 betekent vergroting en een 0 terug naar normaal.

Basic programma

Sprite beweging

```
10 FOR X=832 TO X+62
15 READ Y: POKE X,Y: NEXT
20 POKE 2040,13: POKE 53269,1
25 POKE 53271,1: POKE 53277,1
30 A=53248: B=A+1: X=24: Y=50
40 C=53264: POKE C,0: PRINT CHR$(147)
50 GOSUB 100: POKE 53281,PEEK(56324)
52 POKE 53287,X: GOSUB 500
54 IF Y=210 THEN Y=Y-2: GOTO 60
56 IF PEEK(C)=0 OR X<>50 THEN 52
58 X=X-2: GOTO 80
60 GOSUB 100
62 POKE 53287,X: GOSUB 600
64 IF PEEK(C)=0 OR X<>50 THEN 68
66 X=X-2: GOTO 70
68 IF Y=49 THEN Y=Y+2: GOTO 50
69 GOTO 62
70 GOSUB 100
72 POKE 53287,Y: GOSUB 700
74 IF Y=49 THEN Y=Y+2: GOTO 80
76 IF PEEK(C)=1 OR X<>23 THEN 72
78 X=X+2: GOTO 60
80 GOSUB 100
82 POKE 53287,Y: GOSUB 800
84 IF PEEK(C)=1 OR X<>23 THEN 88
86 X=X+2: GOTO 50
88 IF Y=210 THEN Y=Y-2: GOTO 70
89 GOTO 82
100 POKE 53280,PEEK(56324): RETURN
499 REM*LINKSBOVEN NAAR RECHTSONDER*
500 POKE A,X: POKE B,Y: X=X+1: Y=Y+1
510 IF X=256 THEN X=0: POKE C,1
520 RETURN
599 REM*LINKSONDER NAAR RECHTSBOVEN*
600 POKE A,X: POKE B,Y: X=X+1: Y=Y-1
610 IF X=256 THEN X=0: POKE C,1
620 RETURN
699 REM*RECHTSONDER NAAR LINKSBOVEN*
700 POKE A,X: POKE B,Y: X=X-1: Y=Y-1
710 IF X<0 THEN X=255: POKE C,0
720 RETURN
799 REM*RECHTSBOVEN NAAR LINKSONDER*
800 POKE A,X: POKE B,Y: X=X-1: Y=Y+1
810 IF X<0 THEN X=255: POKE C,0
820 RETURN
899 REM*SPRITEONTWERP TEUTON DOOR INE*
900 DATA0,128,0,0,128,0,0,128
910 DATA0,1,131,192,1,134,0,1
920 DATA254,0,0,120,0,0,216,0
930 DATA192,120,0,128,120,0,193,254
940 DATA0,199,255,128,255,253,192,127
950 DATA252,224,3,255,192,3,103,128
960 DATA15,96,0,12,120,0,12,28
970 DATA0,28,12,0,56,28,0
```

Bijgaand Basic-programma laat een sprite diagonaalsgewijs over het scherm gaan, tot hij tegen een rand botst. Dan maakt hij een hoek van 90 graden met die rand en vervolgt zijn weg.

Positiebepaling.

Zoals al eerder vermeld werd, beweeg je een sprite over het scherm door de coördinaatwaardes in de daarvoor bestemde registers te zetten. Per sprite zijn er twee registers: een voor de horizontale (X) en een voor de vertikale (Y) richting. Register $D010 bevat per Sprite het 9e bit (bit 8) van de X-coördinaat. X-coördinaten kunnen namelijk groter zijn dan 255. Als gewoonlijk hebben bits 0-7 van dit register betrekking op sprites 0-7.

Bepalend voor de plaats van de sprite is het puntje in de linkerbovenhoek. Dit puntje komt op de plaats te staan die aangegeven wordt met de coördinaatwaarde. Welke waarde bij welke plaats op het beeldscherm hoort, is een verhaal apart. We komen hier de volgende keer in een ander verband op terug, maar nu is het voldoende als we weten dat de op het scherm zichtbare coördinaten lopen van 24-249 in de horizontale en van 50-249 in de vertikale richting. Zet je dus 24 in register $D000, 50 in $D001 en maak je bit 0 van $D010 "0", dan zal sprite 0 uiterst linksboven in het scherm verschijnen. Bij kleinere waardes wordt (een gedeelte van) de sprite zichtbaar.

Je kunt het beste zelf wat experimenteren met coördinaatwaardes om te zien wat het effekt is.

Tot zover de sprites in deze aflevering. Volgende keer zullen we ons o.a. bezig gaan houden met Sprite-interrupts en met nog wat andere zaken betreffende de Sprites. Overigens kun je het experimenteren met Sprites wel vergemakkelijken door een goede sprite-editor, die zijn er bv. van Mirrorsoft (Go-Sprite) en een nieuwe van Radarsoft/Topsoft, **Sprite-Machine**, bij dit laatste programma zijn ook animaties erg gemakkelijk te maken. Dat laatste zit overigens in één pakket samen met **Champ**, een assembler/debugger programma. Wie wel deze serie over machinetaal volgt, maar nog geen eigen Assembler heeft om mee te experimenteren, kan daarmee voor weinig geld (*f* 35,-) erg goed uit de voeten.

Analyse

Dan gaan we nu weer verder met de analyse van het **PLOT- programma** uit de vorige aflevering. We zullen ons bezig gaan houden met de gemakkelijkste manier om Basic te kombineren met onze eigen opdrachten.

Elk Basic-programma bestaat uit een aantal opdrachten, die echter niet letterlijk (letter voor letter) in het geheugen staan. De opdrachten die je intypt krijgen elk een eigen code. Dat levert ruimtebesparing op, want zo kun je elk commando met maar 1 byte weergeven, in plaats van de gebruikelijke vier.

De opdracht-codes zijn groter dan 127; dat wil zeggen dat bit 7 altijd is gezet. Dat is gedaan om onderscheid te kunnen maken met de variabelen en cijfers, die wel letterlijk (ASCII) in het geheugen staan. Bij de uitvoering van een programma worden de codes "vertaald" naar startadressen van machinetaalroutines in ROM voor een bepaalde opdracht.
Hieronder is voor een andere methode gekozen m.b.t. de PLOT-opdrachten. Onze opdrachten staan letterlijk in het geheugen. We kunnen dus geen variabelen gebruiken die dezelfde naam hebben als de eigen opdrachten, maar daar staat tegenover dat we het ons een stuk gemakkelijker maken dan de Interpreter het zichzelf doet.

Vektoren

Als we de computer onze eigen opdrachten willen laten uitvoeren, zullen we op de een of andere manier een **wig (wedge)** moeten drijven in het ROM-standaardprogramma van de Interpreter.
Gelukkig heeft de C-64 een aantal standaard-wiggen in huis: de RAM-vectoren. Bij bepaalde (vooral de I/0) opdrachten maakt de Interpreter eerst een indirekte sprong naar het begin van een eigenlijke opdracht. Die sprong gaat dan via twee RAM-geheugenplaatsen welke een vaste pointer vormen, die normaal gesproken wijst naar de instruktie die volgt op de JMP (XXXX). Die RAM-geheugenplaatsen kunnen we laten wijzen naar onze eigen routine, waarmee de **"inbraak"** na de indirekte sprong een feit is.
Wij kunnen de vector op $0308/$0309, die wijst naar de ROM- standaardroutine "voer een opdracht uit", goed gebruiken. Normaal gesproken wijst deze naar $A7E4. Ons programma zal dus beginnen met een routine die de vector verandert. Dat gebeurt vanaf regel $9E2A. We laten hem wijzen naar $9E40, wat een aardig rond hex-getal is. Omdat het PLOT-programma loopt vanaf $9E00 moeten we het beschermen tegen strings, die het O.S. tegen de top van het Basic-geheugen en verder naar beneden zet. We maken de Hi-byte van de Top of Memory-pointer ($0284) dus gelijk aan $9E. Hetzelfde geldt voor de Hi-bytes van de stringbodem-pointer ($0034) en de pointer van de hoogste te gebruiken Basic-lokatie.
Vervolgens keren we terug naar **Basic direct-mode** en gebeurt er pas weer wat, als er een opdracht uitgevoerd moet worden. De Interpreter springt dan met JPM ($0308) naar $9E40 en het eigenlijke PLOT-programma begint. We behandelen deze keer de eerste routine hiervan, die bekijkt of het een van onze opdrachten is. Zo nee, dan springen we weer terug naar de Interpreter, en zo ja, dan voert het programma de opdracht uit.

Commandotest

Het werkt als volgt: de ASCII-codes van onze opdrachten staan van $9E00-$9E1C, gescheiden door $00-bytes. De volgorde is:
CLEAR, TEXT, HIRES, PLOT, UNPLOT.
De beginadressen van de routines voor deze opdrachten staan van $9E1D-$9E26. We vergelijken de opdrachten met waar de Chrget-pointer ($7A) naar wijst en verder. Is het resultaat positief, dan springen we naar de routine voor die bepaalde opdracht. Laten we de testroutine eens stap voor stap bekijken.
(zie illustratie PLOTPROGRAMMA)

dracht gecontroleerd.
Daarvoor wordt X net zo lang verhoogd tot $9E00, X wijst naar de eerstvolgende $00. Dan wordt $02 met 2 vermeerderd om een index te krijgen voor de sprongtabel vanaf $9E1D.
X wordt weer een keer verhoogd en vergeleken met #$1D (alle mogelijke opdrachten getest?).
Is hij gelijk dan wordt er teruggegaan naar Basic: **JMP $A7E4.**
Zo nee, dan wordt de volgende opdracht getest. Het gedeelte G oftewel Gevonden mag de lezer zelf proberen te doorgronden.

```
9E00 43 4C 45 41 52 00 54 45
9E08 58 54 00 48 49 52 45 53
9E10 00 50 4C 4F 54 00 55 4E
9E18 50 4C 4F 54 00 90 9E C0
9E20 9E D8 9E F0 9E F4 9E EA
```

```
9E27 EA          NOP
9E28 EA          NOP
9E29 78          SEI
9E2A A9 40       LDA #$40
9E2C 8D 08 03    STA $0308
9E2F A9 9E       LDA #$9E
9E31 8D 09 03    STA $0309
9E34 8D 84 02    STA $0284
9E37 85 34       STA $34
9E39 85 38       STA $38
9E3B 58          CLI
9E3C 60          RTS
9E3D EA          NOP
9E3E EA          NOP
9E3F EA          NOP
```

PLOT-PROGRAMMA

ROUTINE COMMANDOTEST

```
9E40 A2 00         LDX #$00
9E42 86 02         STX $02
9E44 A0 00      Z  LDY #$00
9E46 C8         S  INY
9E47 B1 7A         LDA ($7A),Y
9E49 C9 20         CMP #$20
9E4B F0 F9         BEQ $9E46        Spaties
9E4D 10 03         BPL $9E52        Karakter
9E4F 4C E4 A7      JMP $A7E4        Basic
9E52 BD 00 9E   K  LDA $9E00,X
9E55 F0 1A         BEQ $9E71        Gevonden
9E57 D1 7A         CMP ($7A),Y
9E59 D0 04         BNE $9E5F        Commando
9E5B C8            INY
9E5C E8            INX
9E5D D0 F3         BNE $9E52        Karakter
9E5F E8         C  INX
9E60 BD 00 9E      LDA $9E00,X
9E63 D0 FA         BNE $9E5F        Commando
9E65 E6 02         INC $02
9E67 E6 02         INC $02
9E69 E8            INX
9E6A E0 1D         CPX #$1D
9E6C D0 D6         BNE $9E44        Zoek
9E6E 4C E4 A7   G  JMP $A7E4
9E71 18            CLC
9E72 98            TYA
9E73 65 7A         ADC $7A
9E75 85 7A         STA $7A
9E77 90 02         BCC $9E7B
9E79 E6 7B         INC $7B
9E7B A6 02         LDX $02
9E7D BD 1D 9E      LDA $9E1D,X
9E80 8D 2E 03      STA $032E
9E83 BD 1E 9E      LDA $9E1E,X
9E86 8D 2F 03      STA $032F
9E89 6C 2E 03      JMP ($032E)
```

De $00 in $02 geeft aan dat we de eerste mogelijke opdracht, **CLEAR,** aan het testen zijn.
Met **Zoek** begint voor elke opdracht de test.
Spaties verplaatst de CHRGET-pointer, Y naar de eerstvolgende niet-spatie. Is de code van dit teken groter dan 127 (bit 7 gezet), dan hebben we met een standaard Basic-opdracht van doen, en wordt er teruggesprongen naar de Interpreter. Zo niet, dan wordt het eerste Chrget-teken het eerste teken van één van onze opdrachten.
Is ons teken een $00, dan betekent dit dat alle voorafgaande tekens de test hebben doorstaan en wordt er uit de testroutine gesprongen naar $9E71.
Is het geen $00, maar zijn de twee tekens wel gelijk, dan wordt het volgende teken getest. Zijn de twee tekens niet gelijk, dan faalt de test en wordt de volgende op-

Vragen en opmerkingen over deze rubriek kun je altijd, liefst schriftelijk, richten aan Commodore Info vanwaar ze naar mij doorgestuurd worden.
Prettige Feestdagen, freaks.!

Beginner kent kracht van aanwinst niet

C-16: meer computer dan je denkt

C-16 special

In de afgelopen jaargang van Commodore Info stond in iedere listingrubriek al wel het één en ander aan programmás, die speciaal voor de C-16 waren bestemd. Daarnaast is er ook een hele serie, meestal kleinere, programma's die met een paar wijzigingen geschikt zijn om op een C-16 te kunnen runnen. Wat er wel en niet mogelijk is met de C-16 is niet met een paar woorden te omschrijven maar hieronder volgen de hoofdzaken.

De meeste C-16 bezitters zijn in het begin nog niet zo ervaren in het zelf programmeren. Veel bezitters zijn zelfs niet op de hoogte van de grote mogelijkheden die in de C-16 aanwezig zijn. Dat geldt uiteraard niet alleen voor de C-16, maar in feite voor alle mensen, die voor het eerst een computer in handen krijgen. De C-16 is echter wel een machine, die vanwege de lage prijzen, waarvoor hij op het moment over de toonbank gaat, veel sneller zal worden gekocht dan de veel duurdere huiscomputers.

Deze populariteit heeft voor- en nadelen. Ten eerste is het zo, dat juist bij deze computer bijzonder weinig aandacht is besteed aan de handleiding en een instruktieboek, waaruit de gebruiker de eerste, nuttige informatie kan halen, om met de C-16 aan de slag te gaan. Zover bekend, zijn er de laatste weken bijna 10.000 nieuwe C-16 computers verkocht, waarvan een groot deel echter zonder enige vorm van documentatie, behalve misschien een garantiebewijs. En zeker bij een gecompliceerd apparaat als de C- 16, is het wenselijk om over wat literatuur te beschikken. De reacties van lezers die bij de redactie aankloppen, wijzen uit, dat er nogal wat mensen zijn, die zo'n machine thuis hebben staan, maar eigenlijk niet weten, wat ze er allemaal mee kunnen doen. Op z'n hoogst hebben ze er een simpel Commodore-spelletje bij, dat dan met moeite van de cassette kan worden geladen, maar daarmee is de pret afgelopen. Kortom, men zit met z'n handen in het haar in plaats van op het toetsenbord.

Vooroordelen

De C-16 is een computer, die tot nog toe is beschouwd als één van de grote mislukkingen van Commodore. En op die manier wordt hij nu ook op grote schaal gedumpt. Maar de kopers zijn feitelijk bijzonder fortuinlijk, omdat ze voor een paar honderd gulden een computer kunnen kopen die wat zijn potentieel betreft vele malen duurder zou moeten zijn. Het enige probleem met de C-16 is het gebrek aan goede software. Voor het overige is het een computer die niet onderdoet voor al zijn voorgangers, met inbegrip van zijn beroemde broer, de C-64.



De C-16 wordt meestal in één adem genoemd met de **Plus/4**. Dat is niet zo gek, als je weet, dat de Plus/4, die overigens officieel niet in Nederland door Commodore op de markt is gebracht, in feite een uitgebreide C-16 is. Alle software die voor de C-16 geschikt is, werkt zonder problemen op de Plus/4. Omgekeerd is dit niet het geval, want er zit minder geheugen in de C-16. Het is mogelijk om ook het geheugen van de C-16 uit te breiden, zodat er intern een echte Plus/4 ontstaat. In het vorige nummer van Commodore Info is daar een artikel aan gewijd. Deze uitbreiding is te koop bij Infolist, Huizen (N.H) tegen een redelijke prijs. De beschikbare 16KByte wordt daardoor uitgebreid tot 64K. Het board wat hiervoor moet worden aangeschaft heet MCT64 en kost rond de 250 gulden.

Bij de aanschaf van de C-16 wordt er helaas te vaak aangenomen dat met deze computer, omdat er Commodore opstaat, alle bestaande software, die voor de C-64 geschreven is, kan worden gerund. Helaas is dit niet waar. Hoewel beide machines nagenoeg dezelfde capaciteiten hebben zijn ze qua ontwerp zó verschillend, dat het een bijna ondoenlijke zaak is, om een C-64 programma over te zetten naar C-16 of Plus/4. Ook met de uitbreiding van RAM tot 64KByte blijft dit hetzelfde. Vergis je daar dus niet in. De telefoongesprekken, waarin wij mensen moeten teleurstellen zijn frequenter dan je je kunt indenken.

Een ander nadeel van de C-16 is dat Commodore speciale pluggen en aansluitingen op de kast heeft gezet. Deze zijn stuk voor stuk van een afwijkend en kostbaar type. De normale datasetterecorder past niet, terwijl de recorders fysiek dezelfde zijn, de gangbare joysticks werken niet en ga zo maar door. Gelukkig komt er schot in de verloop-stukjes en -pluggen. Zo heeft SUZO inmiddels een aanbieding voor haar joysticks, waarbij het mogelijk is om een stekkertje op de normale stick aan te sluiten, zodat deze op de C-16 kan worden gebruikt. Een goede zaak, en we hopen dat de tijd niet ver meer is, dat we voor alle gangbare periferieken kunnen

beschikken over de nodige hard- en software interfacing voor de nieuwste rage in computerland.

Software

Het aanbod van commerciele software in de vorm van spelletjes en serieuze toepassingen is wat mager bij de C-16. Als je op zoek gaat in de computershops, zul je al vrij snel tot de ontdekking komen, dat er bijna niets is, wat je kunt gebruiken op de C- 16. Dat is vooral in het begin een teleurstellende ervaring. Maar nu het erop lijkt, dat de C-16 alsnog een succes gaat worden in Nederland, zal ook hierin wel snel verandering komen. We weten van diverse grote softwarehuizen die bezig zijn om de nodige pakketten uit de Verenigde Staten te importeren. Dus wat dit betreft nog even geduld, en je kunt fantastisch aan de gang. In de tussentijd kun je overwegen om zelf maar eens met de computer aan de slag te gaan in Basic. Echt, programmeren in die taal is niet zo moeilijk. Een voordeel bij de C-16 is de prachtige Basic-interpreter die er standaard in zit. Deze versie is een heel stuk uitgebreider dan die waarover de collega's met een C-64 beschikken. Je kunt bij de C-16 bv. snel en goed cirkels en rechthoeken tekenen en inkleuren op een grafisch 'high-resolution' scherm, terwijl dat bij de C-64 een enorme hoeveelheid POKE en PEEKwerk oplevert. Datzelfde is bij het maken van geluiden en muziek het geval. Elders in dit nummer vind je een paar kleine voorbeelden, die in enkele regels programmatekst al heel leuke effecten op het scherm toveren.
Als je meer wilt weten over Basic, dan kun je het beste ook de Basic Basis cursus, die in dit nummer, begint doorlezen. Deze artikelen zijn niet specifiek voor de C-64 gebruikers bedoeld, dus ook met een C-16 valt daar het één en ander van op te steken. De liefhebbers van machinetaal moeten twee nummers terug beginnen. De nummers 8 en 9 bevatten de eerste delen van een groot artikel over C-16 machinetaal en allerlei handige truuks om hier leuke dingen mee te kunnen doen. Als je je hiervoor interesseert en niet in het bezit bent van deze nummers, zijn ze nog in voorraad bij de uitgever, zodat je ze daar kunt bestellen.

Succes....?

We zijn blij, dat de C-16 zich uiteindelijk een bescheiden plaats heeft weten te verwerven in de range van succesrijke huiscomputers. Ook voor de C-16 geldt het spreekwoord 'Beter laat dan nooit'. We zullen er alles aan doen, om dit ontluikende succes te versterken en in dit blad een vaste plaats reserveren voor deze computers. Daarbij vragen we aan de gebruikers om de zelf geschreven toepassingen en spelletjes voor de C-16 aan ons ter publicatie op te sturen. Ook tips en truuks die door jou worden ontdekt zijn natuurlijk zeer welkom.

Jan Bodzinga. •

Speciale Infolist C-16 cassette

Onze lezersservice Infolist heeft een speciale cassette samengesteld waarop uitsluitend C-16 programma's zijn te vinden.
Alereerst natuurlijk de nieuwe Checksum.C-16 en Visi.C-16, maar daarnaast ook een aantal listings, die in de voorgaande nummers zijn verschenen, listing uit dit nummer en ook nog een aantal niet eerder gepubliceerde listings. Kortom, een waardevolle tape voor beginners én gevorderde C-16 computeraars. Daarnaast zijn de programma's natuurlijk ook allemaal geschikt voor de Plus/4.
De tape is - net als alle andere Infolist produkten - te bestellen door *f* 24,50 (voor de diskette *f* 29,50) over te maken naar giro 3157656 t.n.v. Infolist, Huizen.
Na ontvangst van uw betaling wordt de tape/diskette naar U opgestuurd. Vermeld bij bestellen wel duidelijk: **C-16 tape** of **C-16 diskette.**

Echt aan de gang met de C-16

Checksum en Visilist speciaal aangepast

Tot ons grote genoegen ontdekken steeds meer mensen de specifieke kwaliteiten van de Commodore C-16. In de laatste maanden zijn de verkopen van dit apparaat welhaast groter dan de aantallen die van andere huiscomputers worden verkocht. Voor Commodore-Info is dit de reden, dat we in ons blad meer ruimte zullen gaan toekennen aan software en artikelen die speciaal voor de C-16 zijn geschreven. Jan Bodzinga heeft daarom ook de standaard C-Info intikhulpen aangepast voor de C-16. Voor deze computers (èn de Plus/4) is het vanaf nu mogelijk om gebruik te maken van alle faciliteiten van Commodore-Info, die tot op heden alleen voor de grote groep 64-gebruikers waren weggelegd.

Het aantal gebruikers van de Commodore C-16 computer stijgt met de dag. Hoewel de C-16 al een poos op de markt is, kunnen we nu pas spreken van een succesvolle machine. Dit komt waarschijnlijk, omdat de prijs/prestatieverhouding van het apparaat uiteindelijk ten gunste van de prestaties is uitgevallen. Het is plezierig om dit te kunnen waarnemen. De C-16 is sinds zijn komst op de nederlandse markt zonder meer miskend. Het is wel degelijk een machine waarin een grote hoeveelheid bijzondere kwaliteiten zitten, die we helaas missen bij veel huiscomputers die worden verkocht voor prijzen die meestal meer dan het dubbele van het huidige C-16 prijskaartje bedragen.

Helaas is de support van software en handleidingen sterk achter gebleven bij die van bijvoorbeeld de C-64 of de MSX. Er bestaat relatief erg weinig software voor de C-16. Commodore Info gaat daar wat aan doen. Om te beginnen vind je in deze editie de standaard hulpmiddelen, die we gebruiken om het overtypen en controleren van de Basic-listings gemakkelijk te maken. Dit zijn een paar programma's, die in de loop van de tijd hun nut bewezen hebben voor veel computeraars. We hebben ze aangepast en deels herschreven, zodat ze nu zonder problemen voor de C-16 kunnen worden gebruikt.

Visi.C-16

Dit programma is een prachtig hulpmiddel om Basic-programma's van de disk te kunnen lezen en listen op het scherm of de printer, waarbij door de gebruiker de regel-

lengte en paginagrootte kan worden ingesteld. Daardoor kunnen de listings compleet met alle grafische controletekens leesbaar worden gemaakt.
Alle listings die we in ons blad publiceren zijn op die manier tot stand gekomen.
Om deze Basic-programma's te kunnen overtypen, moet er rekening worden gehouden met een paar dingen. De listing bevat de letterlijke Basic-tekst, die **letterlijk** moet worden ingetypt, zoals die in het blad staat omschreven. Daarnaast vind je de nodige regels, waarin tussen twee rechte haken (spekhaken) bepaalde teksten staan. Deze tekst moet bij het overtypen worden vertaald in controle commando's voor Basic. Dat lijkt moeilijker dan het in werkelijkheid is. Er staat b.v. {4x spatie}. Dit moet worden vertaald in 4 spaties, ofwel 4 maal drukken op de spatiebalk. Bij termen als {op}, {neer} of {links} betreft het de cursor-controletoetsen, die als pijltjes-toetsen op het C-16 toetsenbord aanwezig zijn. Bij {HOME} en {CLR-HOME} moet op de home/clr- toets worden gedrukt. Ook de andere opdrachten tussen deze haken moeten op deze manier worden vertaald en zullen daardoor nog maar weinig echte intik-problemen geven.
Het bekijken van een stuk computer-programmatuur of listing is een heel leerzame methode om iets te gaan begrijpen van het maken van Basic-programma's. De gebruiker kan dan immers in min of meer leesbare taal bekijken welke opdrachten de C-16 tijdens het runnen van zo'n programma moet uitvoeren. Omdat het bekijken van een programma op de monitor niet zo ideaal is, kun je beter een listing op de printer zetten, waardoor er een duidelijk overzicht ontstaat van het hele programma. Lang niet iedere printer kan de speciale Commodore besturingstekens afdrukken, zoals die typische kringels, hartjes en andere bijna niet te ontcijferen tekens. Het programma **Visi.C-16** lost al deze problemen voor u op. Helaas werkt het programma niet met een datasette. Dat komt omdat de C-16, net als de C-64 bij het lezen en schrijven naar de cassetterecorder, de video-chip uitschakelt. Daardoor is het niet mogelijk de standaardopdracht om een Basic-programma sequentieel in te lezen goed te kunnen gebruiken. In de nabije toekomst zullen we proberen ook voor deze groep gebruikers een aanpassing te maken op dit unbieke programma, zodat ook voor hen de weg naar Visi.C-16 wordt geopend.
Sterkte en plezier met dit waardevolle hulpprogramma, waarin alle speciale C-16 en Plus/4 commando's van Basic 3.5 zijn opgenomen. Dit programma kan ook worden gerund op een C-64 of C- 128, maar daarbij moet rekening worden gehouden met de aparte range van C-16 opdrachten, die niet helemaal overeen komen met die van Basic 7.0 uit de C-128.

Checksum.C-16

Checksum.C-16 is de tweede handige hulp, die is afgeleid van de **Syntax.Checksum** die voor de C-64 wordt gebruikt. Deze utility zorgt ervoor, dat na het intypen de regels met een soort **negenproef** worden getest. Per regel krijg je een getal op je scherm. Dit getal moet hetzelfde zijn als het getal wat bij de listing voor dezelfde regel onder het kopje Checksum staat. Is dit niet zo, dan heb je tijdens het intypen een fout gemaakt. Op deze manier is het gemakkelijk om de fouten uit een bepaald programma te halen.
Het in één keer foutloos overtypen van een Basic-programma is bijna niet te doen. Daarom moet je erop zijn bedacht, dat er hier en daar een foutje insluipt. In veel gevallen betekent het echter, dat de C-16 uiteindelijk niet zal doen, wat er in het programma wordt gevraagd. Veelal zal er dan een melding verschijnen op het scherm, waarbij het regelnummer aangeeft waar de fout zit. In andere gevallen is het wat moeilijker om de fout te kunnen opsporen. De C-16 doet wel iets, maar voert niet de juiste opdracht uit. Dan is het een voordeel om deze checksum bij de hand te hebben, zodat de getallen uit het blad kunnen worden vergeleken met die

van het programma dat is overgetypt.

Werkwijze

De manier om met dit programma om te gaan is vrij eenvoudig. Eerst moet het programma letterlijk worden overgetypt in de computer. Elke regel moet beginnen met het intypen van een regelnummer, dat hetzelfde moet zijn als de nummers in het blad. Daarna wordt de tekst van die regel ingetypt. Is het einde van de regel bereikt, dan moet er eerst op de **RETURN-TOETS** worden gedrukt om aan de C-16 door te geven, dat we klaar zijn met deze regel. Gebeurt dit niet, en gaat u bv. met de cursortoetsen naar de volgende regel, dan zal de computer denken dat u nog niet klaar bent met het intypen. De volgende regel wordt dan inclusief een extra regelnummer als het ware aan de vorige geplakt. Dat het programma dan niet goed gaat functioneren spreekt vanzelf.

Als het programma helemaal is ingetypt, moet het worden beveiligd op disk of cassette. Hoe dat gaat kunt u lezen in de handleiding. Voor tape betekent dit, dat u, als u het woord **READY** ziet staan moet typen : **SAVE "CHECKSUM.C-16" <RETURN-TOETS>**

Daarna moet het programma regel voor regel goed worden gecontroleerd. Vooral in de regels die beginnen met het woord **DATA** worden nog al eens fouten gemaakt. Is het hele programma gecontroleerd en zijn de fouten verbeterd dan moet u opnieuw het programma wegschrijven naar cassette of disk. Dit is nodig, omdat tijdens het werken met dit programma de echte gegevens niet meer als Basic, maar als machinetaal in het computergeheugen staan. Als daar ook maar één getal fout is, werkt het hele programma niet meer, en meestal moet de C-16 zelfs opnieuw worden aangezet voordat er weer normaal mee kan worden gewerkt.

Nu ben je klaar om het programma te gebruiken. We doen dit eerst met het programma zelf, dat als Basic-listing immers in het geheugen staat.

U geeft het commando RUN. De computer is even bezig en meldt dan op het scherm: **Data is weggezet. Checksom printen met sys 1536.** Alles wat hierbij nog moet worden gedaan om de getallen op het scherm te krijgen is het intypen van het commando :**SYS 1536 <RETURN>**. Daardoor verschijnen de regelnummers met het testgetal op het scherm. Wilt u de lijst wat langzamer over het scherm laten rollen, dan kan dit, door de **LOGO-toets** ingedrukt te houden. Deze logotoets zit helemaal linksonder op het toetsenbord en is gemerkt met het handelslogo van Commodore en Commodore-Info, de 'gevleugelde' **C**. Als de hele listing van getallen (even) moet worden onderbroken, dan moet er op de **RUN/STOP**-toets worden gedrukt. Zolang deze ingedrukt wordt gehouden, blijft de lijst staan op hetzelfde punt. De C-16 gaat pas weer verder als deze toets wordt losgelaten.

Voor de gelukkigen die een printer bezitten bij hun C-16 het volgende. Het is mogelijk om de lijst met getallen netjes op papier te drukken. Dit vraagt alleen wat meer typewerk. Om te printen moet deze opdrachten worden gegeven:

OPEN 4,4 : CMD 4: SYS 1536 <RETURN>

De printer zal daarna gaan printen. Na de laatste regel moeten er nog twee commando's worden verstuurd om de printer weer af te sluiten. De in te typen regel ziet er daarbij zo uit:

PRINT |4 : CLOSE 4 <RETURN>

De meeste typefouten zullen op deze manier uit een programma kunnen worden verwijderd. Het programma checksum werkt goed, maar heeft als enig nadeel, dat het niet mogelijk is om bv. bepaalde getallen die 'omgedraaid' zijn ingetypt, zoals '243' in plaats van '234'. In de totaaltelling zal bij deze twee getallen de checksum dezelfde zijn.

De werkwijze bij andere programma's die getest moeten worden is gelijk aan de hierboven beschreven handelingen. Alleen moet dan vòòr het testen eerst het programma 'Checksum.C-16' worden geladen en gerund. Daarna typ je **NEW** en laadt het te testen programma. Dit moet niet worden gerund, maar daar moet de op-

dracht **SYS 1536** worden gegeven. De controlegetallen voor het programma verschijnen dan op de monitor. Moeten er nog meer programma's worden getest, dan hoeft niet opnieuw de checksum te worden ingeladen. Deze blijft ongezien in het geheugen van de C- 16 aanwezig, totdat de computer wordt uitgezet.

Veel plezier en hopelijk veel genot van deze twee programma's.

J.B.

Checksum. C-16

```
 10 rem ****************************
    ******
 20 rem syntax.checksum
 30 rem voor c-16 & plus/4
 40 rem
 50 rem syntax testen met 'sys 1536
    '
 60 rem
 70 rem v.851128.16          jan bodzin
    ga
 80 rem ****************************
    ******
 90 i=1536          :rem beginadres
100 reada:ifa>=0then pokei,a:i=i+1:
    goto100
110 print"data is weggezet"
120 print"cheksum printen met 'sys
    1536'
130 end
200 data 165, 43,166, 44,133
210 data  31,134, 32,169,147
220 data  32,210,255,160,  0
230 data 240,  3, 32, 73,  6
240 data  32, 73,  6,208,  1
250 data  96, 72,152, 32,131
260 data   6,168,104,234, 32
270 data  81,  6, 32, 73,  6
280 data 240, 12,201, 32,240
290 data 247, 24,101,252,133
300 data 252, 76, 37,  6,166
310 data 252,169,  0,132,253
320 data  32, 95,164,169, 13
330 data  32,210,255,164,253
340 data  76, 17,  6,200,208
350 data   2,230, 32,177, 31
360 data  96,162,  0,189,123
370 data   6,240,  6, 32,210
380 data 255,232,208,245, 32
390 data  73,  6,170, 32, 73
400 data   6,132,253, 32, 95
410 data 164,162,  3,169, 32
420 data  32,210,255,202,208
430 data 250,169,  0,133,252
440 data 164,253, 96, 82, 69
450 data  71, 69, 76, 32,  0
460 data   0, 72,138, 72, 32
470 data 225,255,240,251,104
480 data 170,104, 96, -1
*** EINDE LISTING ***
```

```
checksum c-16     regel 210  208    regel 360  163
regel 10   249    regel 220  142    regel 370  45
regel 20   247    regel 230  1      regel 380  0
regel 30   121    regel 240  3      regel 390  58
regel 40   143    regel 250  157    regel 400  108
regel 50   75     regel 260  155    regel 410  159
regel 60   143    regel 270  215    regel 420  245
regel 70   8      regel 280  186    regel 430  202
regel 80   249    regel 290  248    regel 440  176
regel 90   103    regel 300  118    regel 450  12
regel 100  2      regel 310  204    regel 460  54
regel 110  245    regel 320  165    regel 470  43
regel 120  237    regel 330  252    regel 480  1
regel 130  128    regel 340  106
regel 200  210    regel 350  98     ready.
```

Visi. C-16

```
  0 rem ********************************
    *****
  5 rem              visi.list
 10 rem           -==============-
 15 rem  listen van basic-programma'
    s
 20 rem kontrole symbolen uitgeprint
    .
 25 rem gebruik: printer & scherm
 30 rem voor c-16 & plus/4
 35 rem
 40 rem programma ontwikkeld door:
 45 rem jan bodzinga, blaricum
 50 rem
 55 rem 851128.5(c16)
 60 rem*******************************
    ****
 99 rem  ----  begin programma  ----
100 printchr$(14)          :rem tekstmod
    e
110 print"[CLR-HOME][2xneer][3xrechts]
    Dit programma gebruikt [RVS-aan]
    DISK-drive[RVS-uit]"
120 gosub2000              :rem maken ar
    rays
130 printtab(5)"[neer][RVS-aan]Filen
    aam[RVS-uit] van te listen[9xspatie]"
140 printtab(5)"programma (op disk)"
    ;:inputpg$
150 kr$="[7xspatie]LISTING programma
     : "+pg$+"[2xspatie]pag."
160 pg$=pg$+",p,r"
170 dv=8:sa=2             :rem device+se
    c.adres
180 printtab(5)"[neer]List op [RVS-aan]
    S[RVS-uit]cherm of [RVS-aan]P[RVS-ui
    t]rinter";:inputp$
190 p$=chr$(asc(left$(p$,1))and127)
200 ifp$<>"s"andp$<>"p"then180
210 ifp$="s"thenps=3:goto280
220 printtab(5)"[neer]Devicenummer p
    rinter ";:inputps
230 printtab(5)"[neer]Aantal regels
    per blad";:inputrg
240 printtab(5)"[neer]Aantal karakte
    rs per regel";:inputkr
250 printtab(5)"[neer][RVS-aan]Print
    er aanzetten / typ >Spatie<"
260 geta$:ifa$<>chr$(32)then260
270 printtab(5)"[op][2xshift-SPATIE]
    >>>>[2xspatie]Printer in orde[5xspat
    ie]<<<<[9xspatie]"
280 ifps=3thenprinttab(5)"[neer]Aant
    al karakters per regel";:inputkr
290 rem ****  initialiseren *******
300 open15,(dv),15  :rem fcharkanaal

310 open2,(dv),(sa),pg$  :rem openen
     programmafile
320 ifps=3thenprint"[CLR-HOME]"   :rem
     clear scherm
330 input#15,en,em$:ifen=0then360
340 printtab(5)"[neer]S O R R Y [RVS-aan
    ]"em$:print
350 close2:close15:goto130
360 rem **** openen print-device *****
370 rl=0:kl=0:te=1:bn=2 :rem tellers
     printer
380 el$="[2xspatie]***[shift-SPATIE]
    EINDE LISTING ***"
390 open4,(ps)
400 print#4,kr$;1:print#4 :rl=3
410 get#2,s1$,s2$          :rem start a
    dres
420 gosub1000:goto440:rem regelnumme
    rs
430 get#2,a$:ifa$=""thengosub1000:if
    notokthen3000
```

```
440  rem *** afhandelen 1 regel *****
     **
450  ifa$=chr$(34)thenquote%=quote%*-
     1:ifnotquote%thengosub600:cr$=""
460  ifnotquote%then520    :rem basict
     okens
470  co$=c$(asc(a$))        :rem karakt
     er
480  if len(cr$)=1 thengosub600: goto
     530
490  if co$<>cr$   thengosub600: goto
     530
500  te=te+1              :rem teller geli
     jke char.
510  goto540
520  print#4,a$(asc(a$));:kl=kl+b%(asc
     (a$)):goto530
530  ifkl>kr-5thenprint#4:gosub1500:print#
     4,s$;:kl=6
540  ifst=0then430
550  :
600  rem **** printen tussen quotes *
     ***
610  iflen(cr$)=1thente=1
620  ifte=1then650
630  te$=str$(te):te$=mid$(te$,2):le=
     len(te$)+2
640  cr$="["+te$+"x"+mid$(cr$,2):te=1
650  ifcr$="[spatie]"thencr$="[shift-SPAT
     IE]"
655  if(len(cr$)>1)and((kl+len(cr$))>
     kr)then700:rem omschr.te lang
660  print#4,cr$;:kl=kl+len(cr$):cr$=
     co$:return
670  :
700  rem *** te lange omschrijving **
     ***
710  qq=kr-kl
720  print#4,left$(cr$,qq):gosub1500:
     print#4,s$;mid$(cr$,qq+1);
```

```
730  kl=6+len(cr$)-qq:cr$=co$
740  return
1000 rem **** afhandelen regelnummer*
     **
1010 ifquote%=-1thengosub600:cr$=""
1020 gosub1500
1030 ok=-1            :rem ok=vlag op
     einde programma
1040 if kl<>6 thenprint#4:goto1070
1050 if ps<>3 thenprint#4,chr$(141);:
     rl=rl-1
1060 if ps =3 thenprint#4,"[7xlinks]"
     ;
1070 get#2,l$:get#2,l$:rem link point
     er
1080 ifl$=""thenprint#4:print#4,el$:o
     k=0:return
1090 get#2,n1$:ifn1$=""thenn1$=chr$(0
     )
1100 get#2,n2$:ifn2$=""thenn2$=chr$(0
     )
1110 r$=str$(asc(n2$)*256+asc(n1$))
     :rem regel-nummer
1120 print#4,left$(s$,6-len(r$))r$" "
     ;:kl=6
1130 get#2,a$            :rem byte bas
     ic programma
1140 quote%=1            :rem geen quot
     emode
1150 return
1160 :
1500 rem ** kontrole op regelaantal *
     **
1510 rl=rl+1:ifps=3then1530
1520 ifrl>rg-3thenrl=3:fori=0to2:print#
     4:next:print#4,kr$;bn:bn=bn+1:print#
     4
1530 return
1540 :
2000 rem******************************
```

```
2010 rem inlezen data in arrays
2020 printtab(5)"[neer][RVS-aan]E e n
     [3xspatie]o g e n b l i k[2xspatie]
     s.v.p.[RVS-uit][2xop]"
2030 dima$(255),c$(255)  :rem a$=basi
     c tokens/c$=karakters + verklaring
2040 dimb%(255)    :rem lengte tokens
2050 fori=0to255:c$(i)=chr$(i):next
     :rem lezen kar.
2060 fori=0to127:a$(i)=chr$(i):next
     :rem lezen tokens
2065 i=128
2070 reada$(i):ifa$(i)<>"-1"theni=i+1
     :goto2070
2080 a$(i)=""
2090 c$(5)="WIT"
2100 c$(17)="neer"
2110 c$(18)="RVS-aan"
2120 c$(19)="HOME"
2130 c$(20)="DELete"
2140 c$(28)="ROOD"
2150 c$(29)="rechts"
2160 c$(30)="GROEN"
2170 c$(31)="BLAUW"
2175 c$(32)="spatie"
2180 c$(129)="oranje"
2190 c$(133)=" F1 "
2200 c$(134)=" F3 "
2210 c$(135)=" F5 "
2220 c$(136)=" F7 "
2230 c$(137)=" F2 "
2240 c$(138)=" F4 "
2250 c$(139)=" F6 "
2260 c$(140)=" F8 "
2270 c$(144)="ZWART"
2280 c$(145)="op"
2290 c$(146)="RVS-uit"
2300 c$(147)="CLR-HOME"
2310 c$(148)="INSert"
2320 c$(149)="bruin"
2330 c$(150)="l.rood"
2340 c$(151)="grijs 1"
2350 c$(152)="grijs 2"
2360 c$(153)="l.groen"
2370 c$(154)="l.blauw"
2380 c$(155)="grijs 3"
2390 c$(156)="PURPER"
2400 c$(157)="links"
2410 c$(158)="GEEL"
2420 c$(159)="CYAAN"
2430 c$(160)="shift-SPATIE"
2435 c$(255)="> PI <"
2440 fori=0to255
2450 b%(i)=len(a$(i))
2460 iflen(c$(i))>1thenc$(i)="["+c$(i
     )+"]"
2480 next
2490 s$="[7xshift-SPATIE]"     :rem re
     gelnummer trailing
2500 return
3000 rem **** sluiten programma *****
3010 print#4:close4
3020 close2:close15
3030 printtab(5)"[neer][RVS-aan]Druk
     toets voor vervolg"
3040 geta$:ifa$=""then3040
3050 print"[CLR-HOME]":goto130 :rem l
     isten volgend prog.
5000 rem******************************
5100 data end,for,next,data,input#,in
     put,dim,read
5110 data let,goto,run,if,restore,gos
     ub,return,rem
5120 data stop,on,wait,load,save,veri
     fy,def,poke
5130 data print#,print,cont,list,clr,
     cmd,sys,open
5140 data close,get,new,tab(,to,fn,sp
     c(,then
5150 data not,step,+,-,*,/,↑,and
5160 data or,>,=,<,sgn,int,abs,usr
5190 data fre,pos,sqr,rnd
5200 data log,exp,cos,sin,tan,atn
5210 data peek,len,str$,val,asc,chr$
5220 data left$,right$,mid$,go,rgr,rclr
5230 data rlum,joy,rdot,dec,hex$,err$
5240 data instr,else,resume,trap,tron
     ,troff
5250 data sound,vol,auto,pudef,graphi
     c,paint
5260 data char,box,circle,gshape,ssha
     pe,draw
5270 data locate,color,scnclr,scale,h
     elp,do
5280 data loop,exit,directory,dsave,d
     load,header
5290 data scratch,collect,copy,rename
     ,backup,delete
5300 data renumber,key,monitor,using,
     until,while
5310 data -1
*** EINDE LISTING ***
```

```
visilist c-16  r 250   59   r 550   58   r 1150  142  r 2210  66   r 3000  253
 r 0   249     r 260   80   r 600   194  r 1160  58   r 2220  69   r 3010  218
 r 5   52      r 270   252  r 610   94   r 1500  142  r 2230  65   r 3020  18
 r 10  63      r 280   111  r 620   73   r 1510  134  r 2240  68   r 3030  177
 r 15  242     r 290   45   r 630   38   r 1520  164  r 2250  71   r 3040  148
 r 20  159     r 300   69   r 640   73   r 1530  142  r 2260  65   r 3050  212
 r 25  222     r 310   52   r 650   174  r 1540  58   r 2270  95   r 5000  123
 r 30  121     r 320   28   r 655   244  r 2000  165  r 2280  231  r 5100  7
 r 35  143     r 330   5    r 660   160  r 2010  39   r 2290  227  r 5110  193
 r 40  159     r 340   169  r 670   58   r 2020  219  r 2300  1    r 5120  241
 r 45  49      r 350   105  r 700   114  r 2030  148  r 2310  160  r 5130  74
 r 50  143     r 360   56   r 710   51   r 2040  54   r 2320  204  r 5140  106
 r 55  38      r 370   159  r 720   193  r 2050  73   r 2330  242  r 5150  198
 r 60  249     r 380   198  r 730   158  r 2060  13   r 2340  245  r 5160  178
 r 99  2       r 390   243  r 740   142  r 2065  150  r 2350  247  r 5190  176
 r 100  143    r 400   232  r 1000  190  r 2070  162  r 2360  60   r 5200  195
 r 110  195    r 410   101  r 1010  127  r 2080  245  r 2370  61   r 5210  59
 r 120  86     r 420   8    r 1020  83   r 2090  87   r 2380  251  r 5220  2
 r 130  157    r 430   13   r 1030  156  r 2100  64   r 2390  40   r 5230  172
 r 140  84     r 440   153  r 1040  186  r 2110  143  r 2400  204  r 5240  228
 r 150  138    r 450   59   r 1050  81   r 2120  65   r 2410  105  r 5250  0
 r 160  16     r 460   35   r 1060  124  r 2130  67   r 2420  57   r 5260  194
 r 170  82     r 470   186  r 1070  118  r 2140  76   r 2430  182  r 5270  127
 r 180  123    r 480   96   r 1080  201  r 2150  226  r 2435  93   r 5280  8
 r 190  136    r 490   131  r 1090  103  r 2160  12   r 2440  236  r 5290  201
 r 200  85     r 500   151  r 1100  106  r 2170  13   r 2450  198  r 5300  91
 r 210  212    r 510   34   r 1110  222  r 2175  217  r 2460  113  r 5310  225
 r 220  180    r 520   184  r 1120  205  r 2180  9    r 2480  130
 r 230  111    r 530   237  r 1130  140  r 2190  60   r 2490  51
 r 240  181    r 540   82   r 1140  49   r 2200  63   r 2500  142  ready.
```

Vanaf het volgende nummer zullen we steeds ruim aandacht aan de C-16 en C-16 listings blijven schenken.